Graph Theory

A NPTEL Course

S.A. Choudum

Department of Mathematics

IIT Madras

Chennai, India

email: sac@iitm.ac.in

Notes to the Reader

- At a faster pace the course can be read in about 65 lectures and at a slower pace in about 100 lectures.

- By skipping certain topics (indicated as optional) the course can be read in about 45 lectures.

- Solve as many exercises as you can. Do not get bogged down on a single exercise for long hours. Revisit the exercises later.

- Those who would like to go for research are advised not to skip any topic.

- A list of the books/monographs is included. These books can be referred to for the proofs which we have not included.

Modules

Module 1 Preliminaries

Contents

1.1 Introduction: Discovery of graphs

We begin the course with a set of problems stated in a quiz format. However, when generalized, these problems lead to deep theorems and interesting applications. So, we urge you to do the following exercises too after solving the quizzes.

- Generalize the problems in various ways.
- Justify your solutions with logical precision.
- Wherever possible design an algorithm to find an optimal solution.
- Code algorithms and run on your desktop. Though coding is not a part of mathematics, this experience will help you to see the solution in a more transparent way.

This whole exercise will enable you to anticipate some of the concepts and theorems before they are stated.

Problem 1: In a college campus, there are seven blocks, Computer Center(C), Library(L), Academic Zone(AC), Administrative Zone(AD), Hospital(H), Guest House(G), Security(S). The problem is to design two LANs satisfying certain conditions:

1. **LAN 1:**
   (i) Two of the blocks are connected to exactly five of the blocks.
   (ii) Two of the blocks are connected to three of the blocks.
   (iii) Three of the blocks are connected to two of the blocks.
2. **LAN 2:**
   (i) Four of the blocks are connected to five of the blocks.
   (ii) Three of the blocks are connected to two of the blocks.

(You can choose the blocks of your choice to satisfy the required conditions. No multiple cables and self loops are permitted.)

Problem 2: There are seven persons. If A calls B, all the information they know is exchanged. Find the minimum number of calls required so that at the end of these many calls, everybody knows the information that everybody else has.

Problem 3: Show that in a party of six persons either (i) there are three persons who know one another or (ii) there are three persons who does not know one another. Can we conclude the same if the party consists of five persons?

Problem 4: A saturated hydrocarbon molecule consists of carbon atoms and hydrogen atoms. Recall from your high school Chemistry that every carbon atom has valency four and every hydrogen atom has valency one. For example, ethane C_2H_6 is such a molecule. If there are n carbon atoms in the molecule, find the number of hydrogen atoms.

Problem 5: In 1996, Nobel prize in Chemistry was awarded to R.F. Curl, R. E. Smalley and H.W. Kroto for their role in the discovery of pure carbon molecules called fullerenes, that is, a fullerene contains carbon atoms and no other atoms. The Kekule structure of a fullerene is a pseudospherical polyhedral shell satisfying the following conditions:

(i) Every corner of the polyhedron is occupied by a carbon atom such that every carbon atom is linked with one other carbon atom with a double bond and with two other carbon atoms with single bonds; notice that this way the valency of every carbon atom is four. (For convenience, the double bonds are replaced by single bonds in creating physical models.)

(ii) Every face of the polyhedron is a pentagonal face or a hexagonal face.

If a fullerene has n carbons atoms, show that $n = 20 + 2k$ for some $k \geq 0$. Thus there is a fullerene with sixty carbon atoms, but no fullerene with sixty nine carbon atoms.

1.2 Graphs

Intuitively, a graph consists of a set of points and a set of lines such that each line joins a pair of points.

• Definitions

- A ***graph*** G is a triple (V, E, I_G), where V, E are sets, and $I_G : E \to V^{(2)}$ is a function, where $V^{(2)} = \binom{V}{2} \cup \{(v, v) : v \in V\}$. An element $\{u, v\} \in \binom{V}{2}$ is denoted by (u, v) or (v, u). For convenience, it is assumed that $V \cap E = \phi$.
- An element of V is called a ***vertex.***
- An element of E is called an ***edge.***
- I_G is called an ***incidence relation.***

Throughout this course, we assume that V and E are ***finite***, and denote $|V|$ by n and $|E|$ by m. V and E are also denoted by V(G) and E(G) respectively.

An example of a graph:

Let $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{a, b, c, d, e, f, g\}$, and let $I_G : E \to V^{(2)}$ be defined as follows: $I_G(a) = (v_1, v_2)$, $I_G(b) = (v_2, v_2)$, $I_G(c) = (v_2, v_3)$, $I_G(d) = (v_3, v_4)$, $I_G(e) = (v_3, v_4)$, $I_G(f) = (v_4, v_5)$, $I_G(g) = (v_1, v_5)$. Then (V, E, I_G) is a graph with five vertices and seven edges. It may be observed that I_G is neither one-one nor onto.

• Pictorial representation of a graph

Any graph is represented by a diagram as follows. Each vertex is represented by a point. If $I_G(x) = (u, v)$, then u and v are joined by a line and it is labeled x. This representation is not unique; you can choose to put the points on the plane wherever you like and draw the lines whichever way you like. Two representations of the above graph are shown in Figure 1.1.

Much of the terminology in graph theory is inspired by such a representation.

Definitions. *Let $G(V, E, I_G)$ be a graph.*

- *If $I_G(x) = (u, v)$, then:*
  (i) x is said to **join** *u and v,*
  (ii) x is said to be **incident** *with u and v and vice-versa,*

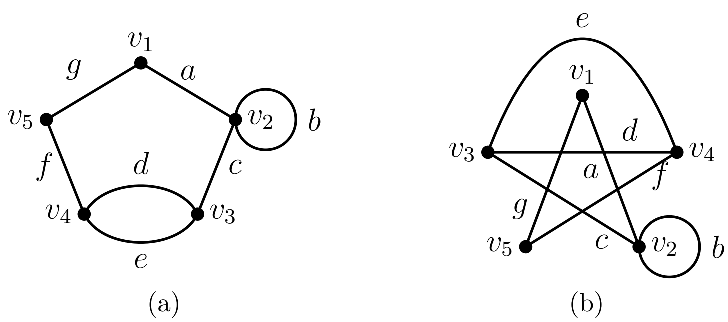


Figure 1.1: Two representations of a graph.

(iii) u and v are said to be **adjacent***,*

(iv) u and v are said to be the **end-vertices** *of x,*

(v) x is denoted by $x(u, v)$, to emphasis its end vertices.

- *If there is no x such that $I_G(x) = (u, v)$, then u and v are said to be* ***non-adjacent.***
- *If x and y are edges incident with a vertex v, then they are said to be* ***adjacent edges****.*
- *If x and y are edges joining the same pair of vertices, then they are called* ***multiple edges****.*
- *If $I_G(x) = (u, u)$, then x is called a* ***loop*** *incident with u.*
- *V is called a* ***simple graph*** *if it has neither multiple edges nor loops. That is, I_G is one-one and $I_G : E \to \binom{V}{2}$.*
  Since I_G is one-one, we can identify any $e \in E$, uniquely with its image $I_G(e) = (u, v) \in \binom{V}{2}$. So, we can alternatively define a simple graph as follows:
- *A* ***simple graph*** *G is a pair (V, E), where V is a non-empty set and $E \subseteq \binom{V}{2}$.*
  For example, consider the graph of Figure 1.1. Here,
  - *v_1 and v_2 are adjacent vertices.*
  - *v_1 and v_3 are non-adjacent vertices.*
  - *a joins v_1 and v_2 and therefore it is incident with v_1 and v_2.*
  - *v_1 and v_2 are the end-vertices of a.*

- *d is not incident with v_1.*
- *b is a loop.*
- *d and e are the multiple edges.*
- *It is not a simple graph.*

A few simple graphs are shown below.

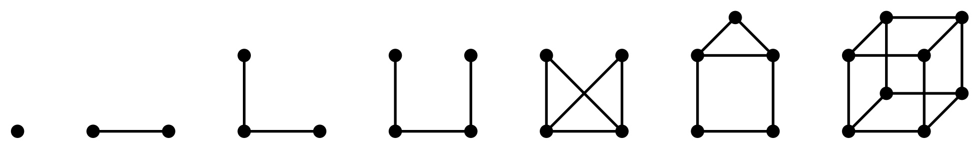


Figure 1.2: Simple graphs

Remark. *While adjacency is a relation among like elements (vertices or edges), incidence is a relation among unlike elements (vertices and edges).*

• Isomorphic graphs

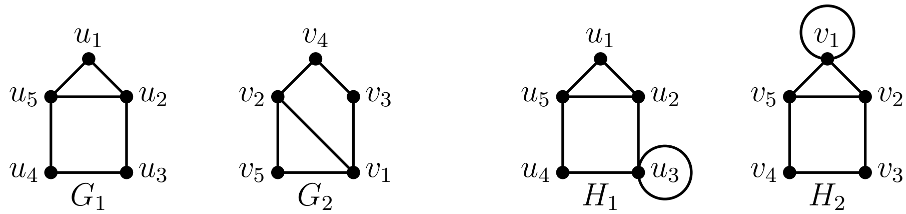
When do we say two graphs are "similar"? The concept of isomorphism is central to all branches of mathematics.

Definitions.

- *Two graphs $G(V, E, I_G)$ and $H(W, F, I_H)$ are said to be* ***isomorphic*** *if there exist two bijections $f : V \to W$ and $g : E \to F$ such that an edge e joins u and v in G if and only if the edge $g(e)$ joins $f(u)$ and $f(v)$ in H.*
- *If G and H are isomorphic, we write $G \simeq H$, and the pair (f, g) is called an* ***isomorphism*** *between G and H.*
- *If $G = H$, then (f, g) is called an* ***automorphism****.*

Figure 1.3 shows a pair of isomorphic graphs and a pair of non-isomorphic graphs.

Remarks.



(a) $G_1 \simeq G_2$ (b) $H_1 \not\simeq H_2$

Figure 1.3: Isomorphic and non-isomorphic graphs.

- Two simple graphs $G(V, E)$, and $H(W, F)$ are isomorphic if and only if there exists a bijection $f : V \to W$ such that two vertices u and v are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H.
- Two graphs $G(V, E, I_G)$ and $H(W, F, I_H)$ are isomorphic if and only if there exists a bijection $f : V \to W$ such that two vertices u and v are joined by k edges in G iff $f(u)$ and $f(v)$ are joined by k edges in H.
- Unfortunately, we have no way to "check" whether two given graphs are isomorphic, except by the brute force method. To show that two given graphs are isomorphic, we have to define a bijection f satisfying the properties stated above. To show that two graphs G_1 and G_2 are non-isomorphic, we have to find a property that G_1 has but G_2 does not have. For example, if $|V(G_1)| \neq |V(G_2)|$ or $|E(G_1)| \neq |E(G_2)|$, then $G_1 \not\simeq G_2$. During this course, we will learn many properties which may be used to show the non-existence of isomorphism.

The problem of designing a "good" (technically, polynomial time) algorithm to check whether two given graphs are isomorphic or not, carries a reward of one million dollars. For details, open any search engine and type "Clay Mathematical Institute".

• Subgraphs

Definition. A graph $H(W, F, I_H)$ is called a ***subgraph*** of $G(V, E, I_G)$, if $W \subseteq V$, $F \subseteq E$ and if $e \in F$ joins u and v in H, then e joins u and v in G (note that the converse is not demanded).

We next define various kinds of subgraphs.

Definitions. *Let $H(W, F, I_H)$ be a subgraph of $G(V, E, I_G)$.*

(i) H is called a ***spanning subgraph****, if $W = V$.*

(ii) H is called an ***induced subgraph*** *if e joins u, v in G, where $u, v \in W$, then e joins u, v in H. H is denoted by $[W]_G$ or $[W]$.*

(iii) If $V_1 \subseteq V$, then $G - V_1$ is the graph $[V - V_1]$. In other words, $G - V_1$ is obtained by deleting every vertex of V_1 and every edge that is incident with a vertex in V_1. In particular, $G - \{v\}$ is denoted by $G - v$.

(iv) If $E_1 \subset E$, then the subgraph $G - E_1$ is obtained by deleting all the edges of E_1 from G. Note that, it is a spanning subgraph of G. If $E_1 = \{e\}$, then $G - \{e\}$ is denoted by $G - e$.

If H is a subgraph of G, we write $H \subseteq G$. If H is an induced subgraph of G, we write $H \sqsubseteq G$.

These various subgraphs are illustrated in Figure 1.4.

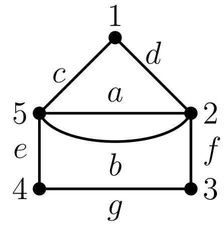
Remarks.

(i) If H is a spanning induced subgraph of G, then $H = G$.

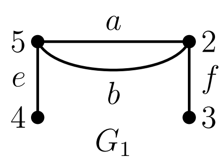
(ii) If H is an induced subgraph of G, then $H = G - V_1$, for some $V_1 \subseteq V$.

• Matrix representations of graphs

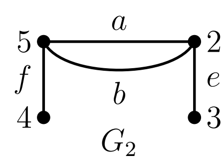
Clearly, we cannot represent a graph pictorially in a computer and hope to compute the parameters associated with graphs. More precisely, pictorial representations do not serve as data structures of graphs. We define below two convenient representations which can serve as data structures.



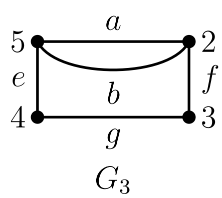
(a) G



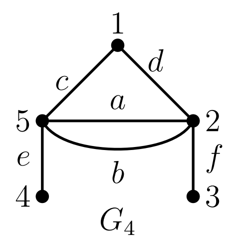
(b) G_1 is a subgraph of G. It is neither an induced subgraph nor a spanning subgraph of G.



(c) G_2 is not a subgraph of G. However, it is isomorphic with a subgraph of G, namely G_1.



(d) G_3 is an induced subgraph of G but it is not a spanning subgraph. In fact, $G_3 = G - 1$.



(e) G_4 is spanning subgraph but not an induced subgraph. In fact, $G_4 = G - g$.

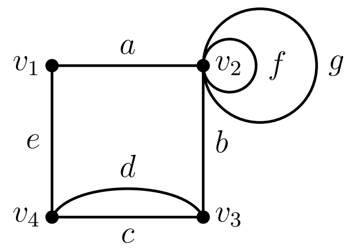
Figure 1.4: A graph G and its various subgraphs.

Definitions. Let G be a graph with vertices $v_1, v_2, \ldots, v_n$ and edges $e_1, e_2, \ldots, e_m$.

(1) The ***adjacency matrix*** $A(G) = [a_{ij}]$ of G, is the $n \times n$ matrix, where

$$a_{ij} = \begin{cases} \text{the number of edges joining } v_i \text{ and } v_j, & \text{if } i \neq j. \\ 2 \times (\text{the number of loops incident with } v_i), & \text{if } i = j. \end{cases}$$

Remarks.



(a) G

	v_1	v_2	v_3	v_4
v_1	0	1	0	1
v_2	1	4	1	0
v_3	0	1	0	2
v_4	1	0	2	0

(b) $A(G)$

	a	b	c	d	e	f	g
v_1	1	0	0	0	1	0	0
v_2	1	1	0	0	0	2	2
v_3	0	1	1	1	0	0	0
v_4	0	0	1	1	1	0	0

(c) $B(G)$

Figure 1.5: A graph G with its adjacency matrix and incidence matrix.

- The adjacency matrix is a symmetric matrix.

- If G is simple, then every entry in $A(G)$ is zero or one and every diagonal entry is zero.

- Any matrix M in which every entry is zero or one is called a ***zero-one*** matrix or a ***binary*** matrix. So, $A(G)$ is a binary matrix if G is simple.

(2) The ***incidence matrix*** $B(G) = [b_{ij}]$ of G, is the $n \times m$ matrix, where

$$b_{ij} = \begin{cases} 0, & \text{if } v_i \text{ is not incident with } e_j. \\ 1, & \text{if } v_i \text{ is incident with } e_j \text{ and } e_j \text{ is not a loop.} \\ 2, & \text{if } v_i \text{ is incident with } e_j \text{ and } e_j \text{ is a loop.} \end{cases}$$

Remark. If G is simple, then B(G) is a zero-one matrix.

• Degree of a vertex

The ***degree*** of a vertex v in a graph G is the number of edges incident with v, with loops counted twice. It is denoted by $deg_G(v)$, $deg(v)$ or simply $d(v)$. The degree of a vertex is also called the ***valency***.

The following equations can be easily observed.

- $\sum_{j=1}^{n} a_{ij} = deg(v_i)$, for every $i, 1 \leq i \leq n$.
- $\sum_{i=1}^{n} a_{ij} = deg(v_j)$, for every $j, 1 \leq j \leq n$.
- $\sum_{i=1}^{n} b_{ij} = 2$, for every $j, 1 \leq j \leq m$.
- $\sum_{j=1}^{m} b_{ij} = deg(v_i)$, for every $i, 1 \leq i \leq n$.

Theorem 1.1. *In any graph G, $\sum_{v \in V(G)} d(v) = 2m$.*

Proof. Every edge contributes 2 to the left hand side sum.

Alternatively, we can also use the incidence matrix $[b_{ij}]$ to prove the theorem. Let $V(G) = \{v_1, v_2, ..., v_n\}$. Then

$$\sum_{i=1}^{n} deg(v_i) = \sum_{i=1}^{n}\sum_{j=1}^{m} b_{ij} = \sum_{j=1}^{m}\left(\sum_{i=1}^{n} b_{ij}\right) = \sum_{j=1}^{m} 2 = 2m. \qquad \square$$

Corollary. *In any graph G, the number of vertices of odd degree is even.*

Proof. Exercise. □

Definitions.

- *If u and v are adjacent vertices in G, then u and v are said to be* ***neighbors****. The set of all neighbors of a vertex x in G is denoted by $N_G(x)$ or $N(x)$. Clearly, if G is simple, then $|N_G(x)| = deg_G(x)$.*

- *If $A, B \subseteq V(G)$ are disjoint non-empty subsets, then*

  $$[A, B] := \{e \in E(G) : e \text{ has one end-vertex in } A \text{ and other end-vertex in } B\}.$$

  If $A = \{a\}$, then $[\{a\}, B]$ is denoted by $[a, B]$.

The following two-way counting argument is often useful in estimating $|[A, B]|$. In this technique, we first look at the edges going out of A and then we look at the edges going out of B; notice that these two sets are equal.

Theorem 1.2. *If $A \subset V(G)$ is a proper subset, then*

$$|[A, V - A]| = \sum_{x \in A} \left(d_G(x) - d_{[A]}(x)\right) = \sum_{y \in V - A} \left(d_G(y) - d_{[V-A]}(y)\right).$$

Proof. (Use of two way counting.) We have

(1) $[A, V - A] = \cup_{x \in A}[x, V - A]$,

(2) $[A, V - A] = \cup_{y \in V - A}[A, y]$,

(3) for any $x \in A$, $d_G(x) = |[x, A - x]| + |[x, V - A]| = d_{[A]}(x) + |[x, V - A]|$, and

(4) for any $y \in V - A$, $d_G(y) = |[y, V - A - y]| + |[y, A]| = d_{[V-A]}(y) + |[A, y]|$.

Therefore,

$$\begin{aligned} |[A, V - A]| &= | \cup_{x \in A} [x, V - A]|, \text{(by (1))} \\ &= \sum_{x \in A} |[x, V - A]| \\ &= \sum_{x \in A} (d_G(x) - d_{[A]}(x)), \text{(by (3))}. \end{aligned}$$

Similarly, (2) and (4) yield

$$|[A, V - A]| = \sum_{y \in V - A} (d_G(y) - d_{[V-A]}(y)).$$

□

Definitions.

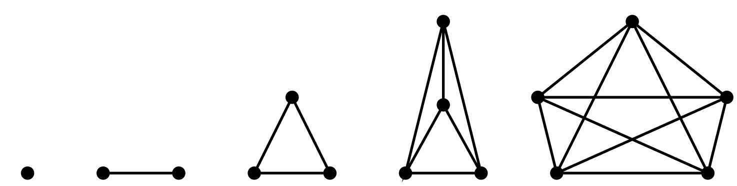
- *A vertex with degree* 0 *is called an* ***isolated vertex***.
- *A vertex of degree* 1 *is called a* ***pendant vertex***.
- *The* ***minimum degree*** *of* G*, denoted by* $\delta(G)$*, is the minimum degree among all the vertices of* G.
- *The* ***maximum degree*** *of* G*, denoted by* $\Delta(G)$*, is the maximum degree among all the vertices of* G.
  Clearly, if G *is a simple graph and* $v \in V(G)$*, then*

$$0 \leq \delta(G) \leq deg(v) \leq \Delta(G) \leq n-1.$$

- *If* $deg(v) = k$*, for every vertex* v *of* G*, then* G *is called a* k***-regular graph***. *It is called a* **regular graph***, if it is* k*-regular for some* k.

• Special graphs

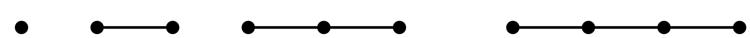
- A simple graph in which any two vertices are adjacent is called a ***complete graph***. A complete graph on n vertices is denoted by K_n. See Figure 1.6.



K_1 K_2 K_3 K_4 K_5

Figure 1.6: Complete graphs.

- A simple graph on n vertices $v_1, v_2, \ldots, v_n$ and $n-1$ edges (v_1, v_2), (v_2, v_3), $\ldots$, (v_{n-1}, v_n) is called a ***path***. It is denoted by $(v_1, v_2, \ldots, v_n)$ or $P(v_1, v_n)$ or P_n. See Figure 1.7



P_1 P_2 P_3 P_4

Figure 1.7: Paths.

- A graph on n distinct vertices $v_1, v_2, \ldots, v_n$ and n edges (v_1, v_2), (v_2, v_3), $\ldots$, (v_{n-1}, v_n), (v_n, v_1) is called a ***cycle.*** It is denoted by $(v_1, v_2, \ldots, v_n, v_1)$ or $C(v_1, v_1)$ or C_n. See Figure 1.8.

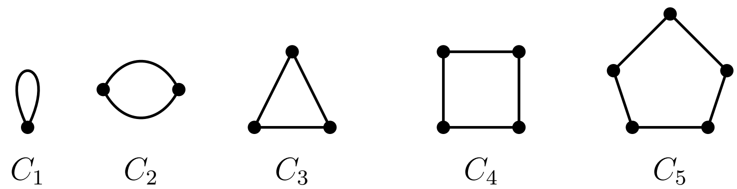


Figure 1.8: Cycles.

- A special graph on ten vertices frequently appears in graph theory. It is called the ***Petersen graph***, after its discoverer J. Petersen (1891). It is shown in Figure 1.9.

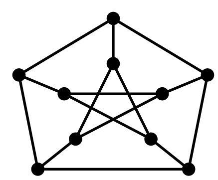


Figure 1.9: Petersen graph.

- A class of regular polyhedra, which have appeared in ancient mathematics are shown in Figure 1.10. These are more popularly called as ***Platonic solids***.

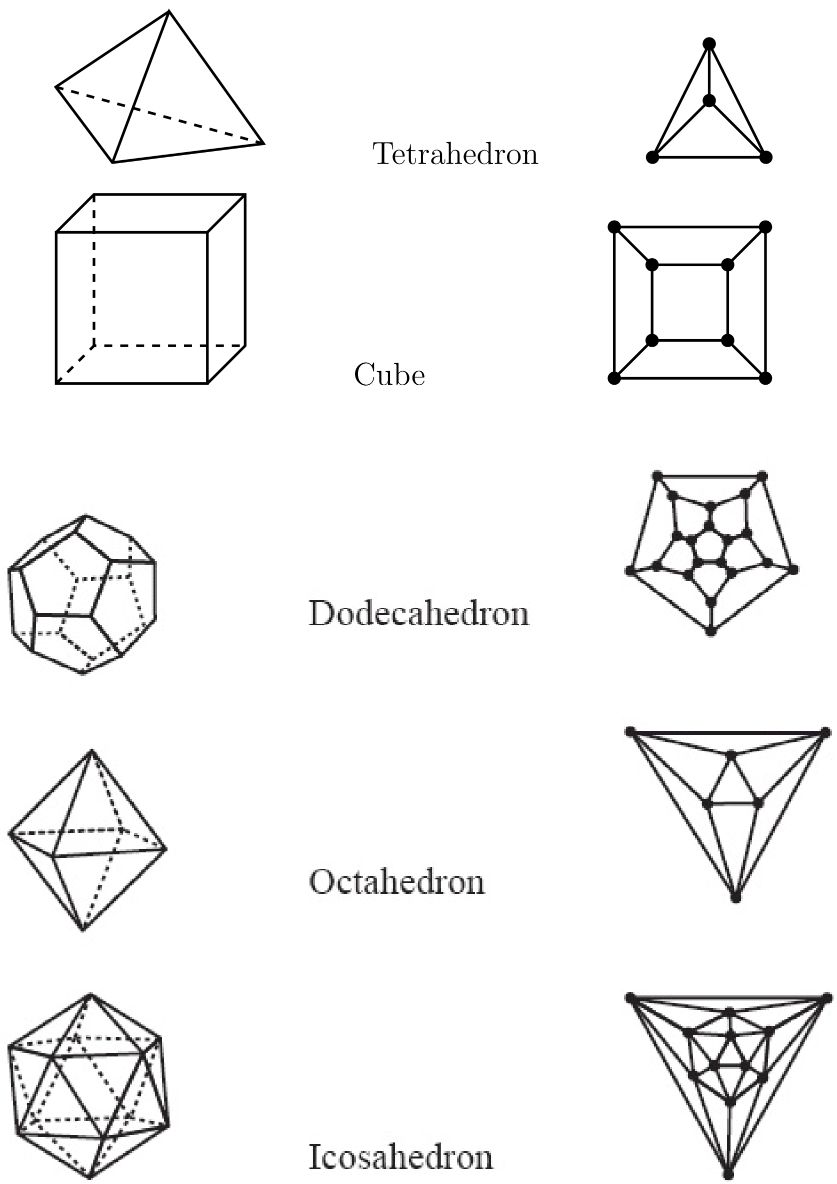


Figure 1.10: Platonic solids and their graphs.

• Complement

- The ***complement*** G^c of a simple graph G has vertex set $V(G)$ and two vertices u, v are adjacent in G^c if and only if they are non-adjacent in G.

- A simple graph G is called a ***self-complementary graph*** if $G \simeq G^c$.

(a) C_4 and its complement (b) Two self-complementary graphs

Figure 1.11: Complement and self-complementary graphs.

• Larger graphs from smaller graphs

We often require an infinite class of graphs with a given property P rather than a single graph. There are several techniques to construct new graphs by "combining" two or more old graphs. These new graphs preserve some of the properties of old graphs. In this subsection we describe a few such techniques.

Let G_1 and G_2 be vertex disjoint graphs with $|V(G_1)| = n_1$, $|E(G_1)| = m_1$, $|V(G_2)| = n_2$ and $|E(G_2)| = m_2$.

Union

The ***union*** of G_1 and G_2 is the graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. It is denoted by $G_1 \cup G_2$. So, $G_1 \cup G_2$ has $n_1 + n_2$ vertices and $m_1 + m_2$ edges.

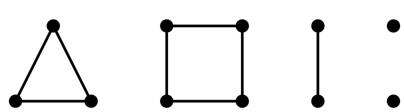
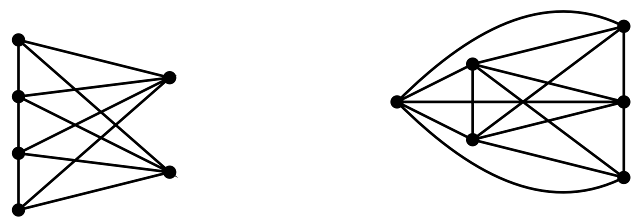


Figure 1.12: Union of graphs: $K_3 \cup C_4 \cup K_2 \cup 2K_1$.

Sum

The ***sum*** or ***join*** of G_1 and G_2 is the graph obtained from $G_1 \cup G_2$ by joining every vertex of G_1 with every vertex of G_2. It is denoted by $G_1 + G_2$. So, $G_1 + G_2$ has $n_1 + n_2$ vertices and $m_1 + m_2 + n_1 n_2$ edges.



(a) $P_4 + K_2^c$. (b) $K_1+K_2+P_3 = K_3+P_3$.

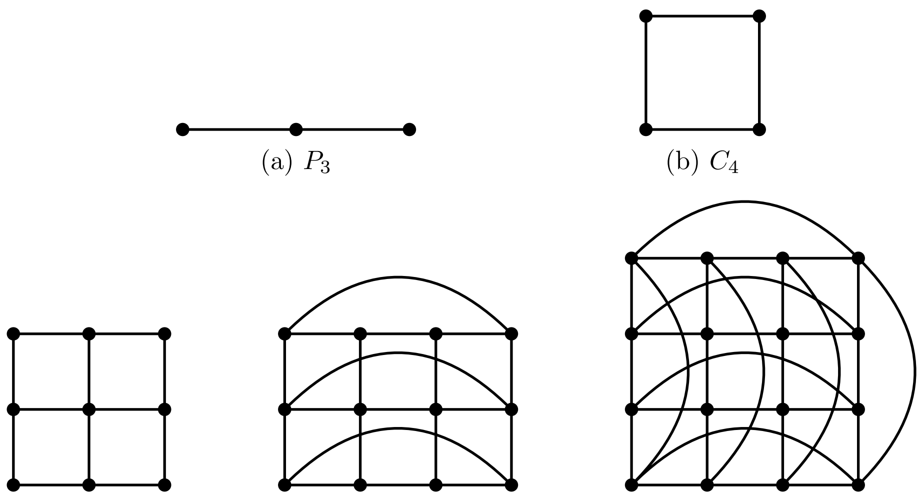
Figure 1.13: Sum of graphs.

The above two definitions can be straightaway extended to the union and the sum of k vertex disjoint graphs $G_1, G_2, \ldots, G_k$. If every $G_i, i = 1, 2, \ldots, k$ is isomorphic with a graph G, then $G_1 \cup G_2 \cup \cdots \cup G_k$ is denoted by kG.

Cartesian Product

The ***Cartesian product*** of simple graphs G_1 and G_2 is the simple graph with vertex set $V(G_1) \times V(G_2)$ in which any two vertices (u_1, v_1) and (u_2, v_2) are adjacent if and only if (i) $u_1 = u_2$, and $(v_1, v_2) \in E(G_2)$ or (ii) $(u_1, u_2) \in E(G_1)$ and $v_1 = v_2$. It is denoted by $G_1 \Box G_2$ or $G_1 \times G_2$. So, $G_1 \Box G_2$ has $n_1 \cdot n_2$ vertices and $n_1 \cdot m_2 + m_1 \cdot n_2$ edges.

The Cartesian product $G_1 \Box G_2 \Box \cdots \Box G_k$ of k simple graphs $G_1, G_2, \ldots, G_k$ has vertex set $V(G_1) \times V(G_2) \times \cdots \times V(G_k)$. Two vertices $(u_1, u_2, \ldots, u_k)$ and $(v_1, v_2, \ldots, v_k)$ are adjacent iff for exactly one i, $1 \leq i \leq k$, $u_i \neq v_i$ and $(u_i, v_i) \in E(G_i)$.



(a) P_3 (b) C_4

(c) $P_3 \Box P_3$ (d) $P_3 \Box C_4$ (e) $C_4 \Box C_4$

Figure 1.14: Cartesian products.

A prime example of the Cartesian product of graphs is the hypercube of dimension d defined by $Q_d = K_2 \Box K_2 \Box \cdots \Box K_2$ (product d times).

Composition

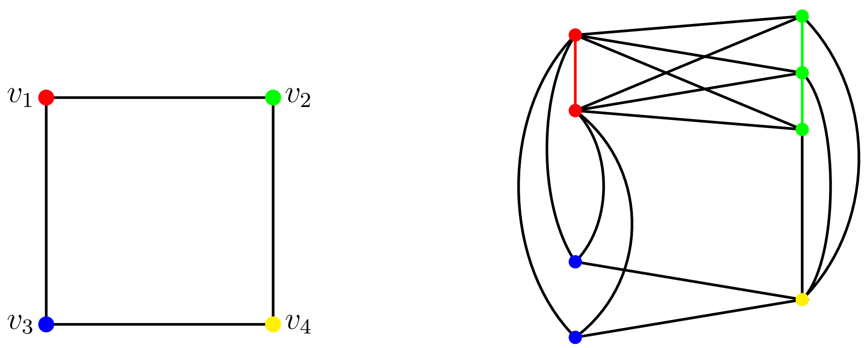
Definition. *Let G have n vertices $v_1, v_2, \ldots, v_n$, and $H_1, H_2, \ldots, H_n$ be any n vertex disjoint graphs. Then the* ***composition*** *$G(H_1, H_2, \ldots, H_n)$ of G with $H_1, H_2, \ldots, H_n$ is the graph obtained as follows:*

(i) Replace each vertex v_i of G by H_i, $i = 1, 2, \ldots, n$. Thus $V(G(H_1, H_2, \ldots, H_n)) = \bigcup_{i=1}^{n} V(H_i)$.

(ii) If v_i and v_j are adjacent in G, then join every vertex of H_i with every vertex of H_j.

(iii) If v_i and v_j are non-adjacent in G, then there is no edge between H_i and H_j.

See Figure 1.15 for an example.



(a) C_4. (b) $C_4(K_2, P_3, K_2^c, K_1)$

Figure 1.15: Composition of graphs.

1.3 Graphic sequences

In this section, we build a graph theoretical model of the LAN problems stated in section 1 and solve them. We recall these problems.

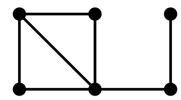
In a college campus, there are seven blocks, Computer Center (C), Library (L), Academic Zone (AC), Administrative Zone (AD), Hospital (H), Guest House (G), Security (S). The problem is to design two LANs satisfying certain conditions:

1. **LAN 1:**
   (i) Two of the blocks are connected to exactly five of the blocks.
   (ii) Two of the blocks are connected to three of the blocks.
   (iii) Three of the blocks are connected to two of the blocks.

2. **LAN 2:**
   (i) Four of the blocks are connected to five of the blocks.
   (ii) Three of the blocks are connected to two of the blocks.

With these problems as motivation, we define the concept of a graphic sequence.

Definition. If G is a graph on n vertices $v_1, v_2, \ldots, v_n$ with degrees $d_1, d_2, \ldots, d_n$ respectively, then the n-tuple $(d_1, d_2, \ldots, d_n)$ is called the ***degree sequence*** of G.



$S : (4, 3, 2, 2, 2, 1)$

Figure 1.16: A graph and its degree sequence.

So every graph G gives rise to a sequence of integers $(d_1, d_2, \ldots, d_n)$. Conversely, we can ask the question: Given a sequence S of integers $(d_1, d_2 \ldots, d_n)$, does there exist a graph G with S as its degree sequence? By Theorem 1.1, one necessary condition for the existence of G is that $\sum_{i=1}^{n} d_i$ is even. It is easy to show that it is also a sufficient condition. However, the question is more difficult if we ask for the existence of a ***simple graph*** G with degree sequence S. Towards this end, we define the following concept.

Definition. *A sequence of non-negative integers $S = (d_1, d_2, \ldots, d_n)$ is said to be **graphic**, if there exists a **simple graph** G with n vertices $v_1, v_2, \ldots, v_n$ such that $deg(v_i) = d_i$, for $i = 1, 2, \ldots, n$. When such a G exists, it is called a **realization** of S.*

• Graph theoretic model of the LAN problem

Problem: Find necessary and sufficient conditions for a sequence $S_n = (d_1, d_2, \ldots, d_n)$ of non-negative integers to be graphic.

This problem leads to the following three problems (and many more).

- Design algorithms to construct a realization of S, if S is graphic.
- When S is graphic, how many non-isomorphic realizations of S are there?
- Given a graph theoretic property P and a sequence of integers $S = (d_1, d_2, \ldots, d_n)$, find necessary and sufficient conditions for the existence of a graph G having the property P and degree sequence $(d_1, d_2, \ldots, d_n)$.

While first two problems have been solved, the third problem is open for many properties P.

We prove two theorems which characterize graphic sequences.

• Havel-Hakimi criterion

Theorem 1.3 (Havel 1955, Hakimi 1962). *A sequence*

$$S : (d_1 \geq d_2 \geq \cdots \geq d_n)$$

of non-negative integers is graphic if and only if the reduced sequence

$$S' : (*, d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n)$$

is graphic.

(Here, S' is obtained from S by deleting d_1 and subtracting 1 from the next d_1 terms. We also assume, without loss of generality, that $d_1 \leq n - 1$.)

Proof.

(1): S' is graphic $\Rightarrow$ S is graphic.

Since S' is graphic, there exists a simple graph G' on $n-1$ vertices $v_2, v_3, \ldots,$ v_n with degrees $d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n$, respectively. Add a new vertex v_1 to G' and join it to $v_2, v_3, \ldots, v_{d_1+1}$. The resultant graph is a realization of S.

(2): S is graphic $\Rightarrow$ S$'$ is graphic.

Since S is graphic, there exists a simple graph G on n vertices $v_1, v_2, \ldots, v_n$ with degrees $d_1, d_2, \ldots, d_n$ respectively. If v_1 is adjacent with $v_2, v_3, \ldots, v_{d_{1+1}}$ then $G - v_1$ is a realization of S'. Else, v_1 is non-adjacent to some vertex v_i, where 2

$\leq i \leq d_1 + 1$. Therefore, v_1 is adjacent to some vertex v_j, where $d_1 + 2 \leq j \leq n$. Since $j > i$, we conclude that $deg(v_i) = d_i \geq d_j = deg(v_j)$. However, v_j is adjacent to v_1 but v_i is not adjacent to v_1. So, there is some v_p such that v_i is adjacent to v_p but v_j is not adjacent to v_p. Hence, G contains the subgraph shown in Figure 1.17.

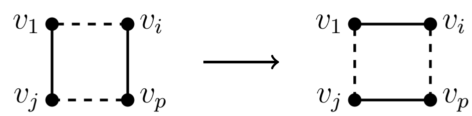


Figure 1.17: Application of 2-switch.

We delete the edges (v_1, v_j), (v_i, v_p) and add the edges (v_1, v_i), (v_j, v_p) (and retain all other edges of G). The resultant graph H is simple and $deg_G(v_k) = deg_H(v_k)$ for every $k, 1 \leq k \leq n$. So, H is also a realization of S in which v_1 is adjacent with ***one more vertex*** in $\{v_2, v_3, \ldots, v_{d_1+1}\}$ than G does. If v_1 is not adjacent to some vertex in $\{v_2, v_3, \ldots, v_{d_1+1}\}$ in H, then we can continue the above procedure to eventually get a realization G^* of S such that v_1 is adjacent to all the vertices in $\{v_2, v_3, \ldots, v_{d_1+1}\}$. Then $G^* - v_1$ is a realization of S. $\square$

• Realization of a graphic sequence

The proof of Havel-Hakimi theorem contains enough information to construct a simple graph with degree sequence $(d_1, d_2, \ldots, d_n)$, if $(d_1, d_2, \ldots, d_n)$ is graphic, else we can use the theorem to declare that $(d_1, d_2, \ldots, d_n)$ is not graphic. We illustrate these remarks by taking the examples of LAN 1 and LAN 2 problems.

LAN 1

Input: $(5, 5, 3, 3, 2, 2, 2)$.

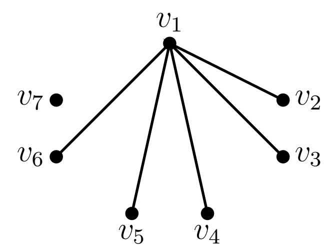
Output: A simple graph G on seven vertices $v_1, v_2, \ldots, v_7$ with degree sequence $(5, 5, 3, 3, 2, 2, 2)$ if the input is graphic, else declaration that the input is not graphic.

Iteration 1:

Input:$(5_1, 5_2, 3_3, 3_4, 2_5, 2_6, 2_7)$. Here, i_j indicates that the degree of v_j will be i in G, at the end of the algorithm, if the input is graphic.

Output:$(*, 4_2, 2_3, 2_4, 1_5, 1_6, 2_7)$

In the figure, we have shown a graph by drawing the vertices $v_1, v_2, \ldots, v_7$ and joining v_1 with v_2, v_3, v_4, v_5, v_6, since we have subtracted 1 from d_2, d_3, d_4, d_5, d_6.

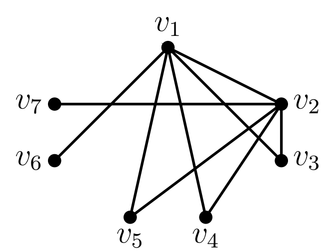


We rearrange this sequence in decreasing order which will be the input for the second iteration.

Iteration 2:

Input: $(*, 4_2, 2_3, 2_4, 2_7, 1_5, 1_6)$

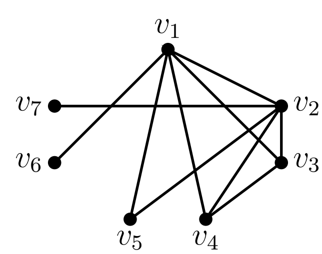
Output: $(*, *, 1_3, 1_4, 1_7, 0_5, 1_6)$



Iteration 3:

Input: $(*, *, 1_3, 1_4, 1_7, 1_6, 0_5)$

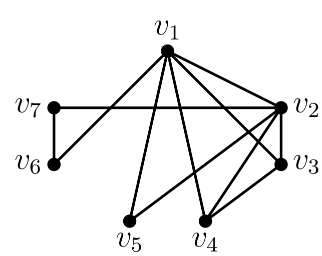
Output: $(*, *, *, 0_4, 1_7, 1_6, 0_5)$



Iteration 4:

Input: $(*, *, *, 1_7, 1_6, 0_4, 0_5)$

Output: $(*, *, *, *, 0_6, 0_4, 0_5)$



The output in the fourth iteration is obviously graphic and so we stop the algorithm. Also, we have realized a graph G with degree sequence (5, 5, 3, 3, 2, 2, 2) shown above.

LAN 2:

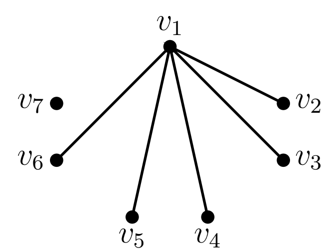
Input: (5, 5, 5, 5, 2, 2, 2)

Output: A graph G with degree sequence (5, 5, 5, 5, 2, 2, 2) or a declaration that the input is not graphic.

Iteration 1:

Input: $(5_1, 5_2, 5_3, 5_4, 2_5, 2_6, 2_7)$

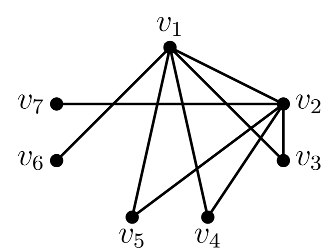
Output: $(*, 4_2, 4_3, 4_4, 1_5, 1_6, 2_7)$



Iteration 2:

Input: $(*, 4_2, 4_3, 4_4, 2_7, 1_5, 1_6)$

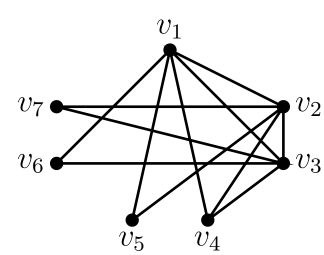
Output: $(*, *, 3_3, 3_4, 1_7, 0_5, 1_6)$



Iteration 3:

Input: $(*, *, 3_3, 3_4, 1_7, 1_6, 0_5)$

Output: $(*, *, *, 2_4, 0_7, 0_6, 0_5)$



We stop the algorithm after the third iteration, since $(2, 0, 0, 0)$ is obviously not graphic. Using Havel-Hakimi Theorem, we declare that the given input is not graphic and hence conclude that the construction of LAN 2 is not possible.

Remarks.

- The realization of a graphic sequence constructed as above is not necessarily unique, since we rearrange the sequence of integers during the iterations. In fact, one may realize two non-isomorphic simple graphs with degree sequence S. See exercise 29.

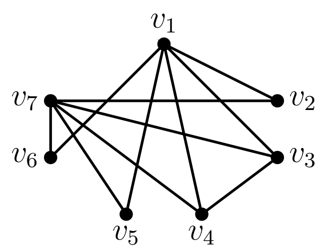


Figure 1.18: Another graph realization of the sequence (5, 5, 3, 3, 2, 2, 2).

- There exist simple graphs which cannot be constructed by the above algorithm. For example, the graph shown in Figure 1.18 cannot be constructed using the Havel-Hakimi algorithm.

• Erdös-Gallai criterion

The next theorem gives an alternative characterization of graphic sequences.

Theorem 1.4 (Erdös and Gallai, 1960)**.** *A sequence $S : (d_1 \geq d_2 \geq \cdots \geq d_n)$ of non-negative integers is graphic if and only if the following hold:*

(EG1) $\sum_{i=1}^{n} d_i$ *is even,*

(EG2) $\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{j=k+1}^{n} \min\{k, d_j\}$, *for every* $k = 1, 2, \ldots, n$.

Proof.

(1) S is graphic $\Rightarrow$ S satisfies (EG1) and (EG2).

Since S is graphic, there exists a simple graph G on n vertices $v_1, v_2, \ldots, v_n$ such that $deg_G(v_i) = d_i$, for $i = 1, 2, \ldots, n$. Therefore, $\sum_{i=1}^{n} d_i = \sum_{i=1}^{n} deg_G(v_i)$ which is an even integer by Theorem 1.1.

Next, we show that (EG2) holds. Let k be an integer such that $1 \leq k \leq n$. Let $A = \{v_1, v_2, \ldots, v_k\}$ and $B = \{v_{k+1}, v_{k+2}, \ldots, v_n\}$. We estimate the maximum value of l.h.s sum $M = \sum_{i=1}^{k} d_i = \sum_{i=1}^{k} deg_G(v_i)$.

(i) Any $v_i \in A$ is adjacent to at most $k-1$ vertices in A. Hence, it contributes at most $k-1$ to M.

(ii) Any $v_j \in B$ is adjacent to all the k vertices in A or d_j vertices in A, whichever is minimum. So it contributes $\min\{k, d_j\}$ to M. Hence, $\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{j=k+1}^{n} \min\{k, d_j\}$.

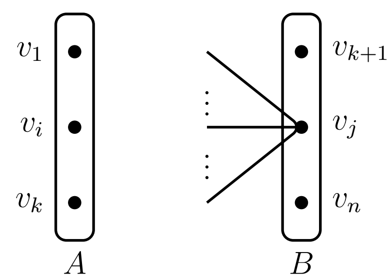


Figure 1.19: $\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{j=k+1}^{n} min\{k, d_j\}$.

(2) (EG1) and (EG2) $\Rightarrow$ S is graphic. We omit the proof. □

We illustrate Erdös-Gallai Theorem by taking LAN 1 and LAN 2 problems.

LAN 1: Here we have to check whether the sequence (5, 5, 3, 3, 2, 2, 2) is graphic.

k	l.h.s	r.h.s		l.h.s $\leq$ r.h.s?
	$d_1+\cdots+d_k$	$k(k-1)$	$\min\{k,d_{k+1}\}+\cdots+\min\{k,d_7\}$	
1	5	0	$1+1+1+1+1+1=6$	✓
2	10	2	$2+2+2+2+2=10$	✓
3	13	6	$3+2+2+2=9$	✓
4	16	12	$2+2+2=6$	✓
5	18	20	$2+2=4$	✓
6	20	30	2	✓
7	22	42	0	✓

Using Erdös-Gallai Theorem we conclude that (5, 5, 3, 3, 2, 2, 2) is graphic.

LAN 2: Here, we have to check whether (5, 5, 5, 5, 2, 2, 2) is graphic.

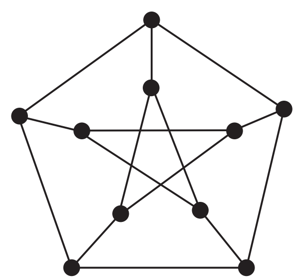
k	l.h.s	r.h.s		l.h.s $\leq$ r.h.s ?
	$d_1+\cdots+d_k$	$k(k-1)$	$\min\{k,d_{k+1}\}+\cdots+\min\{k,d_7\}$	
1	5	0	$1+1+1+1+1+1=6$	✓
2	10	2	$2+2+2+2+2=10$	✓
3	15	6	$3+2+2+2=9$	✓
4	20	12	$2+2+2=6$	×

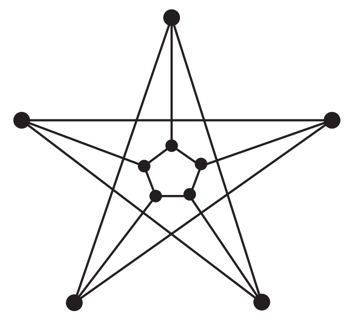
We stop checking, since (EG2) is not satisfied when $k=4$. We conclude that (5, 5, 5, 5, 2, 2, 2) is not graphic.

Remark. Notice that we did not construct a realization of $(5,5,3,3,2,2,2)$. We only verified that the sequence is graphic. So, LAN 1 cannot be solved using Erdös-Gallai Theorem. However, the theorem has good theoretical implications.

Exercises

1. Define an isomorphism between the following two graphs:





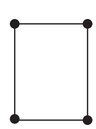
2. (a) If G and H are isomorphic graphs, then show that (i) $|V(G)| = |V(H)|$, (ii) $|E(G)| = |E(H)|$, (iii) if (f, g) is an isomorphism between G and H, then show that $deg_G(v) = deg_H(f(v))$, for every $v \in V(G)$ and (iv) if $f(u) = v$, then $f|_{N(u)}$ is an isomorphism between $[N_G(u)]$ and $[N_H(v)]$.

   (b) Give examples of two non-isomorphic graphs with the same degree sequence.

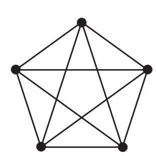
3. Let $u, v \in V(G)$. If there exists an automorphism (f, g) of G, such that $f(u) = v$, then show that $G - u \simeq G - v$.

4. Draw all the non-isomorphic simple graphs on n vertices for $n = 1, 2, 3, 4$.

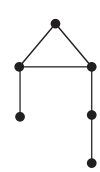
5. Show that the set of all automorphisms of a simple graph G form a permutation group under the usual binary operation of functions. Describe the automorphism groups of the following graphs:



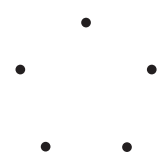
G_1



G_2



G_3



G_4

6. If G is simple, then show that G and G^c have the same automorphism group.

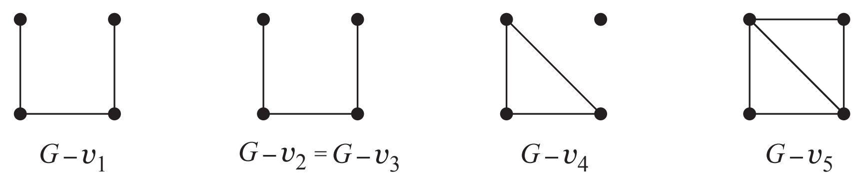
7. If H is a subgraph of a simple graph G, does it follow that H^c is a subgraph of G^c.

8. A simple graph G on 6 vertices $v_1, v_2, v_3, v_4, v_5, v_6$ has (i) 7 edges in $G - v_1$ and $G - v_2$, (ii) 6 edges in $G - v_3$ and $G - v_4$, (iii) 5 edges in $G - v_5$ and $G - v_6$. Find the number of edges in G.

9. Draw all the simple graphs G on 6 vertices such that $G - u \simeq G - v$, for every pair u, v of vertices.

10. A simple graph G on 5 vertices v_1, v_2, v_3, v_4, v_5 is such that (i) $G - v_1 \simeq G - v_2 \simeq G - v_3 \simeq K_2 \cup K_2$, and (ii) $G - v_4 \simeq G - v_5 \simeq K_3 \cup K_1$. Draw G.

11. Let G be a simple graph on v_1, v_2, v_3, v_4, v_5. The graphs $G - v_i, i = 1, 2, 3, 4, 5$ are shown below. Find G.



12. Let G be a simple graph on vertices $v_1, v_2, \ldots, v_n$. Let $G - v_i$ have m_i edges for $i = 1, 2, \ldots, n$. Show the following:
   (i) $m = \frac{1}{n-2} \sum_{i=1}^{n} m_i$,
   (ii) $deg(v_i) = \left(\frac{1}{n-2} \sum_{j=1}^{n} m_j \right) - m_i, i = 1, 2, \ldots, n$.

13. Give an example of a simple graph on 9 vertices and 20 edges which contains no K_3 as a subgraph.

14. Give an example of a graph G on 8 vertices such that neither G contains a K_3 nor G^c contains K_4.

15. (a) Draw a simple graph on 7 vertices with maximum number of edges which contains no complete subgraph on 4 vertices.
   (b) Draw a simple graph on n vertices with maximum number of edges which contains no complete subgraph on p, $2 \leq p \leq n$ vertices.

16. The diagonal entries of the square of the adjacency matrix of a simple graph G are (3, 3, 2, 1, 1). Draw G and its incidence matrix.

17. The adjacency matrix $A(G_1)$ and the incidence matrices $B(G_2)$ and $B(G_3)$ of three graphs G_1, G_2 and G_3 on 5 vertices and 4 edges are shown below. Verify:

(a) Whether G_1 is isomorphic with G_2.

(b) Whether G_2 is isomorphic with G_3.

Justify your answers.

$$\underset{A(G_1)}{\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}} \qquad \underset{B(G_2)}{\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}} \qquad \underset{B(G_3)}{\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}}$$

18. Find the adjacency matrix $A(G)$ and draw the simple graph G whose incidence matrix B is such that

$$BB^T = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 & 1 \\ 1 & 1 & 4 & 1 & 1 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 1 & 1 & 1 & 3 \end{pmatrix}$$

where B^T is the transpose of B.

19. Draw (i) one self-complementary graph on 4 vertices, (ii) two non-isomorphic self-complementary graphs on 5 vertices, and (iii) two non-isomorphic self-complementary graph on 8 vertices.

20. If G is self-complementary then show that $|V(G)| = 4k$ or $4k + 1$, for some k.

21. For each integer n of the form $4k$ or $4k+1$, $k \geq 1$, construct a self-complimentary graph on n vertices using a recursive technique.

22. If G and H are self complimentary graphs, then show that the composition graph $G(H_1, H_2, \ldots, H_n)$ is self complimentary, where $H_i = H$, $i = 1, 2, \ldots, n$.

23. The graph d-cube, denoted by Q_d, is defined as follows:
$V(Q_d) = \{(x_1, x_2, \ldots, x_d) : x_i = 0 \text{ or } 1, 1 \leq i \leq d\}$. Any two vertices $(x_1, x_2, \ldots, x_d)$ and $(y_1, y_2, \ldots, y_d)$ are adjacent in Q_d if and only if $x_i \neq y_i$, for exactly one i, $(1 \leq i \leq d)$. Draw Q_1, Q_2, Q_3. Find (i) $|V(Q_d)|$, (ii) $|E(Q_d)|$, (iii) $deg(x)$, $x = (x_1, x_2, \ldots, x_d) \in Q_d$.

24. Show that $Q_d = K_2 \square K_2 \square \cdots \square K_2$ (Cartesian product d times).

25. Find the minimum integer $k (\geq 1)$ such that there is a simple graph with degree sequence $2^k 4^k 7^k$, where d^k denotes that d is repeated k times.

26. A simple graph G has degree sequence $(d_1, d_2, \ldots, d_n)$. What is the degree sequence of G^c.

27. (a) Verify which of the following sequences are graphic, using (I) Havel-Hakimi Theorem, and using (II) Erdös-Gallai Theorem.
    (i) (5, 5, 5, 2, 2, 2, 1)
    (ii) (4, 4, 4, 4, 2, 2, 0)
    (iii) (7, 6, 5, 4, 4, 3, 2, 1)
    (iv) (5, 5, 4, 4, 2, 2)
    (v) (5, 5, 3, 3, 2, 2)

    (b) Whenever a sequence S is graphic, construct a simple graph with S as degree sequence using Havel-Hakimi Theorem.

28. Given an example of a simple graph that cannot be realized by using the algorithm following Havel-Hakimi criterion.

29. Show that there are only two non-isomorphic realizations of the degree sequence (5, 5, 3, 3, 2, 2, 2).

30. Let $S = (d_1 \geq d_2 \geq \cdots \geq d_n)$ be a sequence of integers. If p $(1 \leq p \leq n)$ is the smallest integer such that $d_p \leq p - 1$, then show that S is graphic iff

    (i) $\sum_i d_i$ is even, and

    (ii) $\sum_{i=1}^k \leq k(k-1) + \sum_{j=p}^n \min\{k, d_j\}$, for every $k = 1, 2, \ldots, p-1$.

31. Show that in any group of two or more persons, there are always two persons with exactly same number of friends.

32. Show that any sequence $(d_1, d_2, \ldots, d_n)$ of non-negative integers is a degree sequence of some graph (not necessarily simple) if and only if $\sum_{i=1}^n d_i$ is even.

33. Show that any sequence $(d_1, d_2, \ldots, d_n)$ of non-negative integers where $d_1 \geq d_2 \cdots \geq d_n$ is a degree sequence of some loopless graph (it can have multiple edges) if and only if (i) $\sum_{i=1}^n d_i$ is even, and (ii) $d_1 \leq d_2 + d_3 + \cdots + d_n$.

34. Give an example of a graphic sequence $(d_1, d_2, \ldots, d_n)$ such that the application of Havel-Hakimi algorithm yields two non-isomorphic graphs. (Choose n as small as possible.)

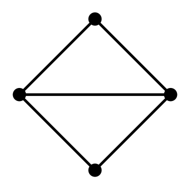
35. Give an example of a simple graph (on as few vertices as you can) which cannot be constructed by Havel-Hakimi algorithm.

36. Let $(4, 4, \ldots, 4, 3, 3, \ldots, 3)$ be sequence of n integers where 4 is repeated k times and 3 is repeated $n - k$ times. Find all the values of k and n for which the sequence is graphic.

37. Let $n \geq 1$ be an integer. Does there exist a simple graph with degree sequence $(n, n, n-1, n-1, \ldots, 3, 3, 2, 2, 1, 1)$? Justify your answer.

38. Show that a regular sequence $(d, d, \ldots, d)$ of length n is graphic if and only if (i) $d \leq n - 1$, (ii) $d \cdot n$ is even.

39. (a) Let G be the graph shown below.



    Construct a 3- regular simple graph H such that $G \sqsubseteq H$.

    (b) Show that any simple graph G has a $\Delta(G)$-regular simple supergraph H such that G is an induced subgraph of H. (If $G \subseteq H$, then H is called a supergraph of G).

Module 2 Connected graphs and shortest paths

Contents

Any network (communication or pipe line or transportation) consists of nodes and physical links connecting certain pairs of nodes. One of the network problems is to move objects (messages/liquids/vehicles) between two given nodes in shortest possible time (or distance).

This real world problem can be easily modeled as a graph theoretic problem. Figure 2.1 shows a communication network with five nodes, each of which is represented by a vertex. An edge represents a direct link. The integer along an edge represents the time to send a message along that link.

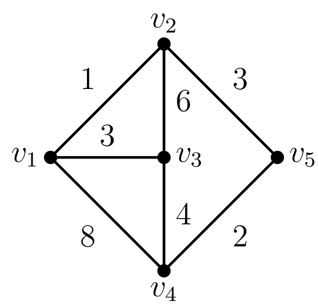


Figure 2.1: A communication network.

By examining all possible paths from v_1 to v_4, we find that the shortest route to send a message from v_1 to v_4 is along the path (v_1, v_2, v_5, v_4) and it takes six units of time. There exist good algorithms to find shortest paths which avoid brute force method of examining all paths between two vertices.

2.1 Walks, trails, paths, cycles

The real world concept of "moving" objects between two nodes is captured in the following terminology.

Definitions. Let G be a graph and let $v_0, v_t \in V(G)$.

- A (v_0, v_t)-***walk*** is a finite alternating sequence

$$W(v_0, v_t) = (v_0, e_1, v_1, e_2, v_2, \dots, e_t, v_t)$$

of vertices and edges such that e_i is an edge incident with vertices v_{i-1} and $v_i, i = 1, 2, \dots, t$.

- v_0 is called the ***origin*** and v_t is called the ***terminus***. Other vertices are called the ***internal vertices***. Note that v_0 and v_t can also be internal vertices.
- The ***length*** of W is the number of edges it contains where an edge is counted as many times as it occurs.
- W is called a ***closed walk***, if $v_0 = v_t$.

Remarks.

- In a walk, vertices and edges may appear any number of times.
- If there exists a (v_0, v_t)-walk, then there exists a (v_t, v_0)-walk.
- If G is a simple graph, W is denoted as a sequence of vertices $(v_0, v_1, \dots, v_t)$ with the understanding that (v_i, v_{i+1}) is an edge, for $i = 0, 1, \dots, t-1$.

Definitions.

- A (v_0, v_t)-walk is called a (v_0, v_t)-***trail***, if no edge is repeated (but vertices may get repeated). It is called a ***closed trail*** if $v_0 = v_t$.
- A (v_0, v_t)-walk is called a (v_0, v_t)-***path***, if no vertex is repeated (and therefore no edge is repeated).

  By definition, every path is a trail and every trail is a walk. However, a walk need not be a trail and a trail need not be a path.
- A closed walk $W(v_0, v_t)$ is called a ***cycle***, if all its vertices are distinct except that $v_0 = v_t$.
- A cycle with k vertices is called a k-cycle and it is denoted by C_k. A C_3 is also a K_3 and it is referred to as a ***triangle***. A 1-cycle is a loop and a 2-cycle consists of two multiple edges.

Any subsequence $W^1(v_i, v_j) = (v_i, e_{i+1}, v_{i+1}, \ldots, v_j)$ of W is called a subwalk of W. We illustrate these concepts by taking a graph.

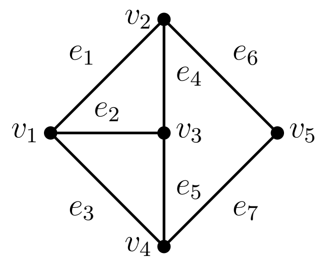


Figure 2.2: A graph G.

Define:

(1) $W_1(v_1, v_5) = (v_1, e_1, v_2, e_4, v_3, e_2, v_1, e_3, v_4, e_5, v_3, e_4, v_2, e_6, v_5)$.

(2) $W_2(v_1, v_5) = (v_1, e_1, v_2, e_4, v_3, e_2, v_1, e_3, v_4, e_7, v_5)$.

(3) $W_3(v_1, v_5) = (v_1, e_1, v_2, e_4, v_3, e_5, v_4, e_7, v_5)$.

(4) $W_4(v_1, v_1) = (v_1, e_1, v_2, e_4, v_3, e_2, v_1, e_3, v_4, e_5, v_3, e_2, v_1)$.

(5) $W_5(v_1, v_1) = (v_1, e_1, v_2, e_4, v_3, e_5, v_4, e_3, v_1)$.

Here, W_1 is a (v_1, v_5)-walk of length 7. It is not a trail. W_2 is a trail of length 5 but it is not a path. W_3 is a path of length 4. W_4 is a closed walk of length 6 but it is not a cycle. W_5 is a cycle of length 4. As per our convention, W_5 is also denoted by $(v_1, v_2, v_3, v_4, v_1)$.

Theorem 2.1. *Every (v_0, v_t)-walk W contains a (v_0, v_t)-path.*

Proof. Among all (v_0, v_t)-subwalks of W, let $P(v_0, v_t)$ be a subwalk of W which has minimum length. We claim that P is a path. Otherwise, there exist i and j (say $i < j$) such that $v_i = v_j$. That is,

$$P(v_0, v_t) = (v_0, \ldots, v_i, e_{i+1}, v_{i+1}, \ldots, v_j(= v_i), e_{j+1}, v_{j+1}, \ldots, v_t).$$

By deleting the subsequence $(e_{i+1}, v_{i+1}, \ldots, v_j)$, we obtain a (v_0, v_t)-subwalk of W which has lesser length, which is a contradiction to the minimality of P. So, $P(v_0, v_t)$ is a path. □

Theorem 2.2. *If G is simple and $\delta(G) \geq 2$, then there exists a cycle of length of at least $\delta(G) + 1$ in G.*

Proof. Let P be a path of maximum length in G. Let $P = (v_1, v_2, \ldots, v_t)$. If v is a vertex adjacent with v_1, then $v \in \{v_2, v_3, \ldots, v_t\}$; else $(v, v_1, v_2, \ldots, v_t)$ is a path of greater length, which is a contradiction to the maximality of P. So, $N(v_1) \subseteq \{v_2, v_3, \ldots, v_t\}$. Let v_k be the last vertex in P to which v_1 is adjacent; see Figure 2.3. Then the subpath $Q = (v_1, v_2, \ldots, v_k)$ contains at least $\deg(v_1) + 1 \geq \delta(G) + 1$

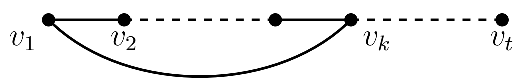


Figure 2.3: A maximum path P(v_1, v_t) and the resultant cycle.

vertices, and so $(v_1, v_2, \ldots, v_k, v_1)$ is a cycle of length $\geq \delta(G) + 1$. □

Theorem 2.3. *Every graph G with $m(G) \geq n(G)$ contains a cycle.*

Proof. If G contains a loop (= a 1-cycle) or a multiple edge (= a 2-cycle) we are through. So, we prove the theorem for simple graphs. This we do by induction on n. If $n \leq 3$, then there is only one graph with $m \geq n$, namely C_3. So we proceed to the induction step. If $\delta(G) \geq 2$, then G contains a cycle by Theorem 2.2. Next assume that $\delta(G) \leq 1$, and let v be a vertex of degree ≤ 1 in G. Then $G - v$ is a graph with $m(G - v) \geq n(G - v)$. Therefore, by induction hypothesis, $G - v$ contains a cycle. Hence G too contains a cycle. □

Definitions. If G contains a cycle, then the following are defined.

- The length of a shortest cycle in G is called its ***girth***.
- The length of a longest cycle in G is called its ***circumference***.

If G is acyclic, then girth and circumference are defined to be ∞.

Graph	P_n	C_n	K_n	$K_m^c + K_n^c$	Q_n	P
Girth	∞	n	3	4	4	5
Circumference	∞	n	n	$2\min\{m,n\}$	2^n	9

Table 2.1: Girths and Circumferences; $m, n \geq 3$ and P is the Petersen graph.

Corollary. *If G is simple and $\delta(G) \geq 2$, then $circumference(G) \geq \delta(G) + 1$.*

Proof. A consequence of Theorem 2.2 □

Theorem 2.4. *If G is a simple graph on least six vertices, then either*

(i) G contains at least three vertices which are mutually adjacent, or

(ii) G contains at least three vertices whcih are mutually non-adjacent.

Proof. Let v be a vertex. Since $n \geq 6$,
either (a) there are at least three vertices, say v_1, v_2, v_3, which are adjacent to v, or (b) there are at least three vertices, say u_1, u_2, u_3 which are non-adjacent to v.

Suppose (a) holds (see Figure 2.4):

If there are two vertices in $\{v_1, v_2, v_3\}$ which are adjacent, say v_1, v_2, then v, v_1, v_2 are three mutually adjacent vertices. On the other hand, if no two vertices of $\{v_1, v_2, v_3\}$ are adjacent, then v_1, v_2, v_3 are three mutually non-adjacent vertices. So, (i) or (ii) holds as claimed in the theorem.

Suppose (b) holds:

If there are two vertices in $\{u_1, u_2, u_3\}$, which are non-adjacent say u_1, u_2, then v, u_1, u_2 are three mutually non-adjacent vertices. On the other hand, if any

Figure 2.4: Adjacency of v.

two vertices in $\{u_1, u_2, u_3\}$ are adjacent, then u_1, u_2, u_3 are three mutually adjacent vertices. So, (i) or (ii) holds. $\square$

The above theorem can be reformulated as follows:

Theorem 2.5. *If G is a simple graph on at least six vertices, then either $K_3 \subseteq G$ or $K_3 \subseteq G^c$.* $\square$

Remarks.

- The assumption $n \geq 6$ made in Theorem 2.4 is necessary. For example, C_5 is a graph on five vertices which satisfies neither (i) nor (ii).
- Which of the problems stated at the beginning of this course is now solved?

2.2 Connected graphs

Clearly, we can move objects between two nodes if they are "connected".

Definitions.

- In a graph G, two vertices u and v are said to be ***connected***, if there exists a (u, v)-path.
- G is said to be a ***connected graph*** if any two vertices are connected; else, G is said to be a ***disconnected graph***.
- A maximal connected subgraph H of a graph G is called a ***component*** of G; maximal in the sense that if H_1 is a connected subgraph of G such that $H \subseteq H_1$, then $H = H_1$.

- The number of components in a graph G is denoted by $c(G)$. So, $c(G) = 1$ if and only if G is connected.

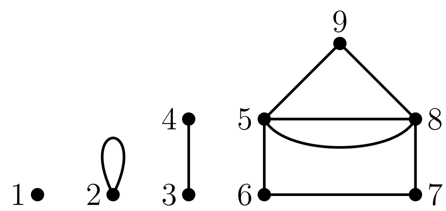
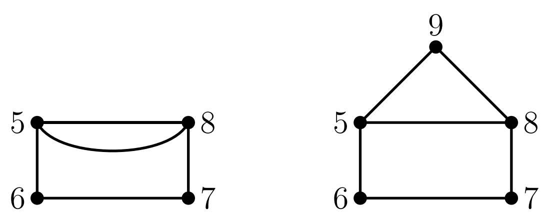


Figure 2.5: A graph G with four components.



(a) H_1 (b) H_2

Figure 2.6: Two subgraphs of G which are not its components; they are connected subgraphs but not ***maximal*** connected subgraphs.

Theorem 2.6. *For any graph G, $m(G) \geq n(G) - c(G)$.*

Proof. We prove the theorem by induction on n.

Basic step: If $n = 1$, the result is obvious.

Induction step: If $\delta(G) \geq 2$, then $m(G) = \frac{1}{2}\left(\sum_{v \in V(G)} d(v)\right) \geq \frac{1}{2}(2n(G)) = n(G) > n(G) - c(G)$. If $\delta(G) = 0$, let v be a vertex of degree 0. Then

$$\begin{aligned} m(G) &= m(G - v) \\ &\geq n(G - v) - c(G - v), \text{ by induction hypothesis,} \\ &= (n(G) - 1) + (c(G) - 1) \\ &= n(G) - c(G). \end{aligned}$$

Next assume that $\delta(G) = 1$, and let v be a vertex of degree 1. Then $c(G-v) = c(G)$. So,

$$\begin{aligned} m(G) &= m(G-v)+1 \\ &\geq n(G-v) - c(G-v) + 1, \text{ by induction hypothesis,} \\ &= (n(G)-1) - c(G-v) + 1 \\ &= n(G) - c(G). \end{aligned}$$ □

Corollary. *For any connected graph G, $m(G) \geq n(G)-1$.* □

Theorem 2.7. *Every graph with $m \geq n-1$ is either connected or contains a cycle.*

Proof. It is enough if we prove the theorem for simple graphs. Assume the contrary and let G be a simple graph with $m \geq n-1$ which is neither connected nor contains a cycle. Let $G_1, G_2, \ldots, G_t$ be its component where $t \geq 2$. Let G_i have n_i vertices and m_i edges, $i = 1, 2, \ldots, t$. Since G_i is acyclic, using Theorem 2.3, we deduce that $m_i \leq n_i - 1$. So,

$$\begin{aligned} m &= m_1 + m_2 + \cdots + m_t \\ &\leq (n_1 - 1) + (n_2 - 1) + (n_t - 1) \\ &= n - t \\ &\leq n - 2. \end{aligned}$$

This is a contradiction to our assumption that $m \geq n-1$. □

Theorem 2.8. *A connected simple graph G contains a cycle if and only if $m \geq n$.*

Proof. If $m \geq n$, then G contains a cycle by Theorem 2.3. The reverse implication can be proved by induction on n by following the proof of Theorem 2.7. □

The above theorem characterizes the graphs with a cycle. However, it is a difficult open problem to find sufficient conditions (or necessary conditions) for the existence of a cycle C_k of specified length k. The following theorem gives a sufficient condition for a simple graph to contain a C_3.

Theorem 2.9. *Every simple graph with $m > \frac{n^2}{4}$ contains a cycle of length* 3(= *triangle*).

Proof. We prove that if G is a simple graph which has no triangles, then $m \leq \frac{n^2}{4}$. This we do by induction n.

Case 1: n is even.

If $n = 2$, then the inequality is obvious. So, we proceed to the induction step assuming that G has $n + 2$ vertices and that the theorem holds for all graphs on n vertices. There exists an edge (u, v) in G; else, $m = 0$. We partition E(G) into three parts as follows:

(1) $E(G) = E(G - \{u, v\}) \cup \{e \in E(G) :$ e has one end vertex in $\{u, v\}$ and other end vertex in $G - \{u, v\}\} \cup \{(u, v)\}$.

Since $G - \{u, v\}$ has no triangles, $m(G - \{u, v\}) \leq \frac{n^2}{4}$, by induction hypothesis. No vertex in $G - \{u, v\}$ is adjacent to both the vertices u and v, since G has no triangles. So

$$|\{e \in E(G) : e \text{ has one end vertex in } \{u, v\} \text{ and other end vertex in } G - \{u, v\}\}|$$
$$\leq n(G - \{u, v\}) = n.$$

Hence, $m(G) \leq m(G - \{u, v\}) + n + 1 \leq \frac{n^2}{4} + n + 1 = \frac{(n+2)^2}{4}$.

Case 2: n is odd. Proof is exactly as above. □

• Distance

The concept of "distance" occurs in every branch of mathematics; graph theory is no exception.

Definitions. Let G be a graph and $u, v \in V(G)$.

- The ***distance*** $d_G(u, v)$ or $d(u, v)$ between u and v is defined as follows:

$$d_G(u,v) = \begin{cases} \text{length of a shortest } (u,v) \text{ path}, & \text{if } u \text{ and } v \text{ are connected}, \\ \infty, & \text{if } u \text{ and } v \text{ are not connected}. \end{cases}$$

- The ***diameter*** of G, $\mathrm{diam}(G)$, is defined by

$$diam(G) = \begin{cases} \max\{d(u,v) : u, v \in V(G)\}, & \text{if } G \text{ is connected}, \\ \infty, & \text{if } G \text{ is disconnected}. \end{cases}$$

If G is connected, then the function $d : V(G) \times V(G) \to \mathbb{R}$ is a metric on $V(G)$. Formally, we state this fact as a theorem. Its proof is easy and hence it is left as an exercise.

Theorem 2.10. *If G is a connected graph, then $(V(G), d)$ is a metric space. That is:*

(i) $d(u, v) \geq 0$, *for every* $u, v \in V(G)$.
(ii) $d(u, v) = 0$ *iff* $u = v$.
(iii) $d(u, v) = d(v, u)$, *for every* $u, v \in V(G)$.
(iv) $d(u, v) \leq d(u, w) + d(w, v)$, *for every* $u, v, w \in V(G)$. □

Theorem 2.11. *Let $A = [a_{ij}]$ be the adjacency matrix of a simple graph G. Then the (i,j) th entry $[A^p]_{ij}$ in A^p is the number of walks of length p from v_i to v_j.*

Proof. We prove the theorem by induction on p. If $p = 1$, then A^p is A and the result is obvious. Suppose that the result is true for $p = r$ and let $p = r + 1$. We have,

$$[A^{r+1}]_{ij} = \sum_{k=1}^{n} [A^r]_{ik}\ a_{kj}.$$

Now

$$[A^r]_{ik}\ a_{kj} = \begin{cases} [A^r]_{ik}, & \text{if } a_{kj} = 1, \text{ that is, if } v_k \text{ and } v_j \text{ are adjacent.} \\ 0 & \text{if } a_{kj} = 0, \text{ that is, if } v_k \text{ and } v_j \text{ are not adjacent.} \end{cases}$$

By induction, $[A^r]_{ik}$ is the number of walks of length r connecting v_i and v_k. Each of these walks is extendable to a walk of length $r+1$ connecting v_i and v_j iff v_k and v_j are adjacent, that is, iff $a_{kj} = 1$. So, by the above equations, $[A^{r+1}]_{ij}$ is the number of walks of length $r+1$ connecting v_i and v_j. □

Theorem 2.12. *If G is a connected simple graph, then the distance between v_i and v_j is the smallest integer $p(\geq 0)$ such that $[A^p]_{ij} \neq 0$.*

Proof. By the minimality of p, $[A^r]_{ij} = 0$, for every $r, 0 \leq r \leq p-1$. So, by Theorem 2.11, there is no walk of length $\leq p-1$ connecting v_i and v_j; hence, $d(v_i, v_j) \geq p$. Since $[A^p]_{ij} \neq 0$, there does exist a walk, say $W(v_i, v_j)$, from v_i to v_j. Since, $[A^r]_{ij} = 0$, for all r, $0 \leq r \leq p-1$, $W(v_i, v_j)$ is a walk of minimum length and hence it is a path. So, $d(v_i, v_j) \leq p$. Combining the two inequalities we get $d(v_i, v_j) = p$. □

• Cut-vertices and cut-edges

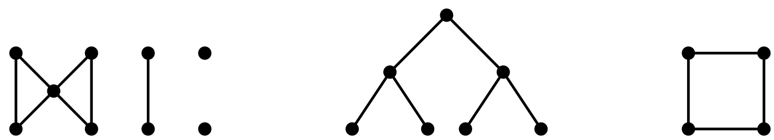
The following two concepts make precise the notions of faulty nodes and faulty links. At the outset, observe that the number of components in $G - v$ may increase or decrease or remain the same, when compared to the number of components in G. In fact, $c(G - v) < c(G)$ iff $deg(v) = 0$.

Let G be a graph, v be a vertex and e be an edge.

Definition. *v is said to be a **cut-vertex** if $c(G - v) > c(G)$.*

Remarks.

- If G is connected, then v is a cut-vertex if $G - v$ is disconnected.
- v is a cut-vertex of G if and only if v is a cut-vertex of a component of G.



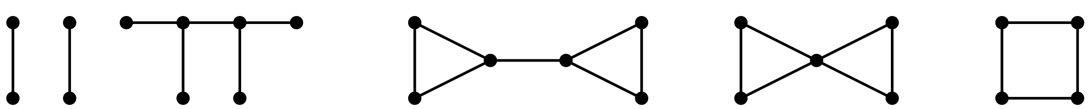
(a) G_1 (b) G_2 (c) G_3

Figure 2.7: G_1 contains exactly one cut-vertex, G_2 contains three cut-vertices and G_3 contains no cut-vertices.

Definition. *e is said to be a cut-edge of G if $c(G - e) > c(G)$.*

Remarks.

- If $e(u, v)$ is a cut-edge of G, then u and v are in two different components of $G - e$. Moreover, $c(G - e) = c(G) + 1$.
- The two remarks made above with respect to a cut-vertex hold good for a cut-edge also.



(a) G_1 (b) G_2 (c) G_3 (d) G_4

Figure 2.8: Every edge in G_1 is a cut-edge, G_2 contains exactly one cut-edge and two cut-vertices, G_3 contains a cut-vertex but contains no cut-edge, G_4 contains neither a cut-vertex nor a cut edge.

Theorem 2.13. *A vertex v of a connected graph is a cut-vertex if and only if there exist vertices x and y $(\neq v)$ such that every (x, y)-path contains v.*

Proof. (1) Suppose v is a cut-vertex. $G - v$ is disconnected so it contains at least two components, say C and D. Let $x \in V(C)$ and $y \in V(D)$. Since there is no (x, y)-path in $G - v$, it follows that every (x, y)-path in G contains v.
(2) Suppose there exist x and y ($\neq v$) such that every (x, y)-path contains v. It follows that there is no (x, y)-path in $G - v$. Hence $G - v$ is disconnected, that is v is a cut-vertex. □

Theorem 2.14. *Let G be a connected graph with at least three vertices. If $e(u, v)$ is a cut-edge in G, then either u or v is a cut-vertex.*

Proof. In $G - e$, there exist two components C and D such that $u \in V(C)$ and $v \in V(D)$. Since $n \geq 3$, either $n(C) \geq 2$ or $n(D) \geq 2$, say $n(C) \geq 2$. Then u is a cut-vertex of G. □

Remarks.

- By the above theorem, it follows that if G is connected, $n(G) \geq 3$ and G has a cut-edge, then G has a cut-vertex. However, the converse is false: The graph G_3 shown in Figure. 2.8 contains a cut-vertex but contains no cut-edges.
- If $e(u, v)$ is a cut-edge, then it is not necessary that both u ***and*** v are cut-vertices; if $d(u) \geq 2$ and $d(v) \geq 2$, then both u and v are cut-vertices.

Theorem 2.15. *An edge $e(u, v)$ is not a cut-edge of G if and only if e belongs to a cycle in G.*

Proof. (1) Suppose e is not a cut-edge.

So, $G - e$ is connected, and u, v are in the same components of $G - e$. So, there exists a path $P(u, v)$ in $G - e$. But then, $(P(u, v), v, u)$ is a cycle in G containing e.
(2) Suppose e belongs to cycle C in G.

So, $C - e$ is a (u, v)-path in $G - e$. Hence, u and v are in the same components of $G - e$, that is $G - e$ is connected. Therefore, e is not a cut-edge. $\square$

Theorem 2.16. *Every connected graph G with $n \geq 2$, contains at least 2 vertices which are not cut-vertices.*

Proof. Let P be a path of maximum length in G; let $P = P(x, y)$. Our claim is that neither x nor y is a cut-vertex of G. On the contrary, suppose x is a cut-vertex and consider $G - x$; see Figure 2.9. It contains at least 2 components say C and D where $y \in D$. Let v be any vertex in C. Then every (v, y)-path in G contains x.

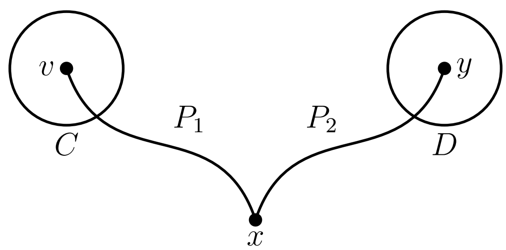


Figure 2.9: Existence of two paths P_1 and P_2 in G.

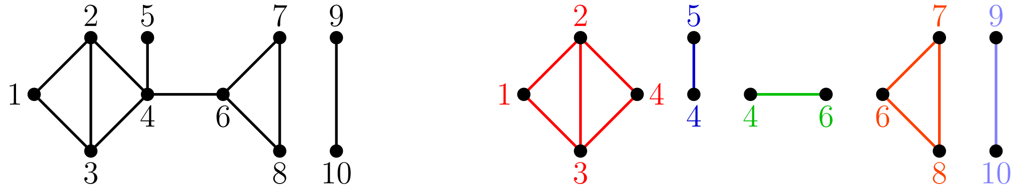
But then $Q = (P_1(v, x), P_2(x, y))$ is a path of length greater than the length of P, a contradiction to the maximality of P. $\square$

• Blocks

Definition. A maximal connected subgraph B of a graph G that has a no cut-vertex of its own is called a ***block*** of G; maximal in the sense that if H is a connected subgraph of G that has no cut-vertex and $H \subseteq B$, then $H = B$; see Figure 2.10.

If G has no cut-vertices, then G is called a block.

Remarks.



(a) G (b) Blocks of G.

Figure 2.10: A graph G and its five blocks.

- A block of a graph does not have a cut-vertex of its own. However, it may contain cut-vertices of the whole graph. In example 2.10a, the edge joining 4 and 6 is a block and both the end points are cut-vertices of the whole graph.
- By definition, G itself is a block if it is connected and it has no cut-vertex.
- Two blocks in a graph share at most one vertex; else, the two blocks together form a block, thus contradicting the maximality of blocks.
- If two distinct blocks of G share a vertex v, then v is a cut-vertex of G.
- Any two distinct blocks are edge disjoint; so the edge sets of blocks partition the edge set of G.
- To establish a property P of a graph G, often it is enough to establish P for each of its blocks, and thereby simplify the proofs.

Definition. Two paths $P(x, y)$ and $Q(x, y)$ are said to be ***internally disjoint*** if they have no internal vertex common.

The following result is a special case of a theorem proved by Menger (1932).

Theorem 2.17. *Let G be a graph with $n(G) \geq 3$. Then G is a block if and only if given any two vertices x and y of G, there exist at least two internally disjoint (x, y)-paths in G.*

Proof. (1) Given any two vertices x, y of G, there exist at least two internally disjoint (x, y)-paths $\Rightarrow G$ is a block.

By the hypothesis, G is connected. So, we have to only show that G has no cut-vertices. On the contrary, suppose v is a cut-vertex of G. Then by Theorem 2.13, there exist vertices x, y such that every (x, y)-path passes through v. This implies that there do not exist two internally disjoint (x, y)-paths, which is a contradiction to the hypothesis.

(2) G is a block $\Rightarrow$ Given any two vertices x, y of G, there exist at least two internally disjoint (x, y)-paths.

We prove the implication by induction on the distance $d(x, y)$.

Basic step: $d(x, y) = 1$. Let e be an edge joining x and y. Then $G - e$ is connected; else, x or y is a cut-vertex which is a contradiction, since G is a block. Hence, there exists a path $P(x, y)$ in $G - e$. But then (x, e, y) and $P(x, y)$ are two internally disjoint (x, y)-paths.

Induction step: $d(x, y) > 1$. Let $P(x, y)$ be a shortest (x, y)-path; so length of $P = d(x, y)$. Let w be a vertex that precedes y in P. So, $d(x, w) = d(x, y) - 1$. Hence, by induction hypothesis, there exist two internally disjoint (x, w)-paths, say $Q(x, w)$ and $R(x, w)$; see Figure 2.11.

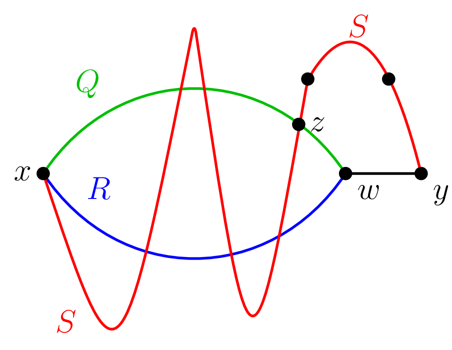


Figure 2.11: Construction of two internally disjoint (x, y)-paths in G.

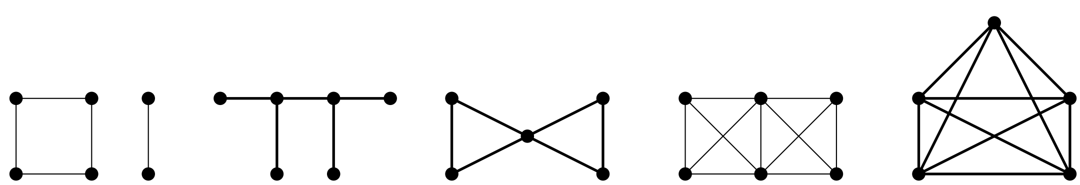
Since G is a block, w is not a cut-vertex. So, there exists a path $S(x, y)$ in $G - w$. Let z be the last vertex in $S(x, y)$ that lies in $V(Q) \cup V(R)$. For definiteness, let $z \in V(Q)$. Then $(Q(x, z), S(z, y))$ and $(R(x, w), w, y)$ are two internally disjoint (x, y)-paths. □

A reformulation of the above theorem yields an alternative characterization of a block.

Corollary. *A graph G with at least three vertices is a block if and only if given any two vertices x and y, there exists a cycle containing x and y.* □

2.3 Connectivity

Consider the graphs shown in Figure 2.12 representing communication or transportation networks. They are successively more robust. k-vertex-connectivity and k-edge-connectivity are two basic parameters that measure the robustness of a graph/network.



(a) G_1 (b) G_2 (c) G_3 (d) G_4 (e) G_5

Figure 2.12: Successively more robust graphs.

Definitions (k-vertex-connectivity)**.**

- A vertex subset $W \subseteq V(G)$ is called a ***vertex-cut*** if $G - W$ is disconnected or $G - W$ is a single vertex graph.
- The integer

$$k_0(G) = \min\{|W| : W \subset V(G), W \text{ is a vertex-cut}\}.$$

is called the ***vertex-connectivity*** of G. That is, $k_0(G)$ is the minimum number of vertices whose deletion disconnects G or results in a graph with a single vertex.

- Any vertex-cut $K \subseteq V(G)$ such that $|K| = k_0(G)$ is called a ***minimum vertex-cut*** of G.
- A graph G is said to be p-***vertex-connected***, if $k_0(G) \geq p$.

Remarks.

- $k_0(G) = 0$ if and only if G is disconnected or G is a single vertex graph.
- $k_0(G) = 1$ if and only if G is connected and it has a cut-vertex.
- $k_0(G) \geq 2$ if and only if G is a block.
- $k_0(G) = n(G) - 1$ if and only if $G \supseteq K_n$.
- A graph G may have many minimum vertex-cuts but $k_0(G)$ is unique.
- If G is p-vertex-connected, then it is t-vertex-connected for every t, $1 \leq t \leq p$. Thus a 3-connected graph is also a 2-connected graph and a connected graph.

Definitions (k-edge-connectivity)**.**

- An edge subset $F \subseteq E(G)$ is called an ***edge-cut*** if $G - F$ is disconnected or $G - F$ is a single vertex graph.
- The integer

$$k_1(G) = \min\{|F| : F \subseteq E(G) \text{ and } F \text{ is an edge cut}\}$$

is called the ***edge-connectivity*** of G. That is, $k_1(G)$ is the minimum number of edges whose deletion disconnects G or results in a single vertex graph.

- Any edge-cut $F \subseteq E(G)$ such that $|F| = k_1(G)$ is called a ***minimum edge-cut***.
- A graph G is said to be s-***edge-connected*** if $k_1(G) \geq s$.

Remarks.

- $k_1(G) = 0$ if and only if G is disconnected or G is a single vertex graph.
- $k_1(G) = 1$ if and only if G is connected and G has a cut-edge.
- $k_1(G) \leq \delta(G)$; this follows since by deleting all the edges incident with a vertex of minimum degree, we disconnect the graph.
- A graph G may have many minimum edge-cuts but $k_1(G)$ is unique.
- If G is s-edge-connected, then it is t-edge-connected, for every t, $1 \leq t \leq s$.
- If F is a minimum edge-cut, then $G - F$ contains exactly two components A and B such that $[V(A), V(B)] = F$.

Table 2.2 shows the above two parameters for the graphs shown in Figure 2.12, some standard graphs ($n \geq 3$) and the Petersen graph P.

	G_1	G_2	G_3	G_4	G_5	K_n	P_n	C_n	P
k_0	0	1	1	2	4	$n-1$	1	2	3
k_1	0	1	2	3	4	$n-1$	1	2	3

Table 2.2: Vertex and edge connectivity of graphs

Theorem 2.18. *For any graph G on at least two vertices, $k_0(G) \leq k_1(G) \leq \delta(G)$.*

Proof. We have already remarked above that $k_1(G) \leq \delta(G)$. Below we prove that $k_0(G) \leq k_1(G)$. If G is disconnected or G is a single vertex graph, then $k_0(G) = 0 = k_1(G)$, $\delta(G) \geq 0$. So, next assume that G is connected. If $n(G) = 2$, then $k_0(G) = 1$, $k_1(G) =$ number of edges joining the two vertices, and $\delta(G) \geq$ number of edges joining the two vertices . Therefore the result follows. Next assume that G is connected and $n(G) \geq 3$.

Case 1: G is simple.

Let $k_1(G) = \lambda$ and let $F = \{e_1(u_1, v_1), \ldots, e_\lambda(u_\lambda, v_\lambda)\}$ be a minimum edge-cut where u_i and v_j need not be distinct. Let $H = G - (\{u_1, u_2, \ldots, u_{\lambda-1}\} - \{u_\lambda, v_\lambda\})$. Then e_λ is a cut-edge of H. So, u_λ or v_λ is a cut-vertex of H; say u_λ. But then

$\{u_1, \ldots, u_{\lambda-1}, u_\lambda\}$ is a vertex-cut of G. Therefore, $k_0(G) \leq \lambda = k_1(G)$.

Case 2: G is any graph.

Let H be an underlying simple spanning subgraph of G. Then $k_0(H) = k_0(G)$ and $k_1(H) \leq k_1(G)$. We deduce that $k_0(G) = k_0(H) \leq k_1(H) \leq k_1(G)$. □

Remark. The inequalities shown in Theorem 2.18 are best possible in the following sense. Given any three integers r, s, t such that $0 \leq r \leq s \leq t$, there exists a simple graph G with $k_0(G) = r$, $k_1(G) = s$, $\delta(G) = t$. (We leave it as an exercise to construct such a graph G. You can take a hint from the two graphs shown in Figure 2.13.)

(a) G_1 (b) G_2

Figure 2.13: $k_0(G_1) = 1$, $k_1(G_1) = 2$, $\delta(G_1) = 3$; $k_0(G_2) = 3$, $k_1(G_2) = 4$, $\delta(G_2) = 4$.

Two central theorems on connectivity are due to K. Menger (1932).

Theorem 2.19. *A graph G is k-vertex connected* $(1 \leq k \leq n-1)$ *if and only if given any two distinct vertices u and v, there exist k internally disjoint (u, v)-paths (that is, no two of the paths have an internal vertex common).*

Theorem 2.20. *A graph G is k-edge connected if and only if given any two distinct vertices u and v, there exists k edge-disjoint (u, v)-paths (that is, no two paths have an edge common).*

There are many sufficient conditions for a graph to be k-vertex-connected or k-edge-connected. We state and prove two results, and a few are included in the exercise list.

Theorem 2.21. *Let $1 \leq k \leq n$. If $deg_G(v) \geq \left\lceil \frac{n+k-2}{2} \right\rceil$, for every vertex v, then G is k-vertex-connected.*

Proof. Let $W \subseteq V(G)$ be a k-vertex-cut. Then $G-W$ contains at least two components and so it has a component D such that $d := |V(D)| \leq \frac{n-k}{2}$. Let x be a vertex in D; see Figure 2.14.

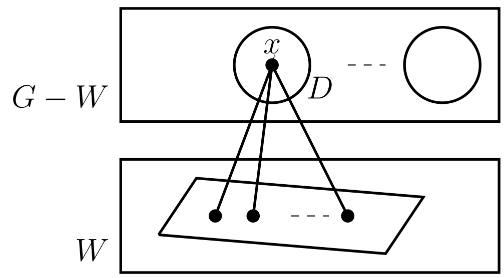


Figure 2.14: Estimate for $deg_G(x)$.

Then x can be adjacent to at most $d-1$ vertices in $G-W$ and k vertices in W. So, $deg_G(x) \leq d-1+k \leq \frac{n-k}{2} - 1 + k = \frac{n+k-2}{2}$, which is a contradiction to the hypothesis. $\square$

If G is simple and $diam(G) = 1$, then $G = K_n$ and so $k_1(G) = n-1 = \delta(G)$. The next results shows that simple graphs with diameter 2 also attain maximum possible edge-connectivity.

Theorem 2.22 (J. Plesink, 1975)**.** *If G is simple and $diam(G) = 2$, then $k_1(G) = \delta(G)$.*

Proof. **(Technique of two-way counting)** Let F be a a minimum edge-cut; so $|F| = k_1(G)$. Then G contains exactly two components, say $[A]$ and $[B]$ such that $[A, B] = F$. Let $|A| = a$ and $|B| = b$.

If there exists some pair of vertices $x \in A$ and $y \in B$ such that $[x, B] = \emptyset$ and $[A, y] = \emptyset$, then $\text{dist}(x, y) \geq 3$, contrary to the hypothesis. So, $[x, B] \neq \emptyset$, for every $x \in A$ or $[A, y] \neq \emptyset$, for every $y \in B$. For definiteness, let $[x, B] \neq \emptyset$, for every $x \in A$.

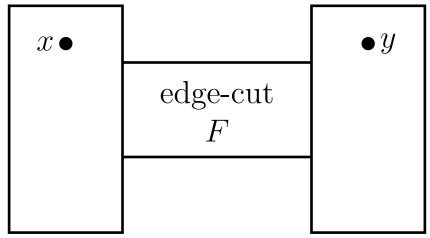


Figure 2.15: Two components $[A]$ and $[B]$ of $G - F$.

Therefore,

$$k_1 = |F| = \sum_{x \in A} |[x, B]| \geq \sum_{x \in A} 1 = a. \tag{2.1}$$

We next estimate $|F|$, using the following equation and deduce that $k_1 \geq \delta$.

$$d_G(x) = d_{[A]}(x) + |[x, B]|, \tag{2.2}$$

$$\begin{aligned}
k_1 = |F| &= \sum_{x \in A} |[x, B]| \\
&= \sum_{x \in A} d_G(x) - d_{[A]}(x), \text{by } (2.2) \\
&\geq \delta a - a(a-1), \text{ since } d_{[A]}(x) \leq a - 1, \text{ for every } x \in A \\
&= \delta + \delta(a-1) - a(a-1) \\
&= \delta + (a-1)(\delta - a) \\
&\geq \delta + (a-1)(k_1 - a), \text{ since } \delta \geq k_1, \text{ by Theorem 2.18} \\
&\geq \delta, \text{ since } a \geq 1, \text{ and } k_1 \geq a, \text{ by } (2.1).
\end{aligned}$$

Since $k_1 \leq \delta$, for every graph (Theorem 2.18), the result follows. □

2.4 Weighted graphs and shortest paths

In any network, the links are associated with “band widths” or “lengths” or “capacities”. In discrete mathematics all these parameters are called the “weights” of the links.

• Weighted graphs

Definitions.

- A pair $(G, \mathcal{W})$, where G is a graph and $\mathcal{W} : E(G) \to \mathbb{R}$ is any function is called a ***weighted graph***; $\mathcal{W}(e)$ is called the weight of e. We assume that $\mathcal{W}(e) \geq 0$, for every edge e.
- Convention: Any unweighted graph G is treated as a weighted graph with $\mathcal{W}(e)$ $= 1$, for every edge e.
- If H is a subgraph of G, then its weight $\mathcal{W}(H)$ is defined to be $\sum_{e\in E(H)} \mathcal{W}(e)$. So, the weight of a path P in G is $\sum_{e\in E(P)} \mathcal{W}(e)$. It is called the *weighted length* of P.
- The ***weighted distance*** $dist(u, v)$ between two given vertices u and v in a weighted graph $(G, \mathcal{W})$ is defined as

$$dist(u,v) = \begin{cases} \min\{\mathcal{W}(P) : \ P \text{ is a } (u,v)\text{-path}\}, & \text{if } u \text{ and } v \text{ are connected,} \\ \infty, & \text{if } u \text{ and } v \text{ are not connected.} \end{cases}$$

- We often drop the adjective “weighted” to reduce the writing.

  In the graph of Figure 2.16:

  - The weight of G is 14.5.
  - The weight of the path (u, a, v, c, x, e, y) is 12.
  - The weight of the path (u, a, v, d, x, e, y) is 4.5.
  - The weight of the path (u, b, x, e, y) is 5.

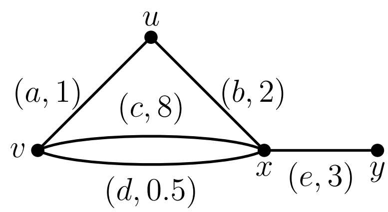


Figure 2.16: A weighted graph $(G, \mathcal{W})$, where the weight of an edge is shown next to its label.

- The shortest (u, y)-path is (u, a, v, d, x, e, y); so $d(u, y) = 4.5$; however, in the underlying "unweighted" graph, $d(u, y) = 2$.

If G is a connected weighted graph, define $d : V(G) \times V(G) \to \mathbb{R}$ by $d(u, v) = dist(u, v)$, for every $u, v \in V(G)$.

Theorem 2.23. *If G is a connected weighted graph with non-negative edge weights, then the following hold.*

1. $d(u, v) \geq 0$, for every u, $v \in V(G)$.

2. $d(u, v) = 0$, if and only if $u = v$ or $\mathcal{W}(u, v) = 0$.

3. $d(u, v) = d(v, u)$, for every $u, v \in V(G)$.

4. $d(u, v) \leq d(u, x) + d(x, v)$, for every $u, v, x \in V(G)$.

Proof. Exercise. □

In many real world network problems, one is often required to find the shortest distance and a shortest path between two given nodes. In graph theoretic terminology, these two problems can be modeled as follows:

Design an algorithm to find the shortest distance and a shortest path between two given vertices in a given weighted graph.

We describe two well-known such algorithms. It is assumed that the edge weights are non-negative.

• Dijkstra's shortest path algorithm (1959)

It is a "one-to-all" algorithm in the sense that given any vertex $u_0 \in V(G)$, the algorithm outputs the distance between u_0 and every other vertex. However, it can be easily modified into an "all-to-all" algorithm or to a "many-to-many" algorithm. It is based on the principle that if $(u_0, e_1, u_1, e_2, u_2, \ldots, u_{k-1}, e_k, u_k)$ is a shortest (u_0, u_k)-path, then $(u_0, e_1, u_1, e_2, u_2, \ldots, u_{k-1})$ is a shortest (u_0, u_{k-1})-path.

◦ **Description of the algorithm**

Input: A weighted graph $(G, \mathcal{W})$ and a vertex $u_0 \in V(G)$.

If a pair of vertices u and v are non-adjacent it is assumed that they are joined by an edge of large weight, say $2^{\mathcal{W}(G)}$, denoted ∞.

Output: The weighted distance $d(u_0, x)$, for every $x \in V(G)$.

Remark.

- ◦ It is a labeling process where every vertex x is dynamically labeled $l(x)$ and at the termination of the algorithm, $l(x)$ indicates $d(u_0, x)$.
- ◦ Each vertex x is associated with an array of 3 fields $\boxed{x \mid l(x) \mid p(x)}$, where $p(x)$ is the parents of x.

Step 1: Initialization

$l(u_0) \leftarrow 0; p(u_0) = u_0; l(x) \leftarrow \infty$ and $p(x) = NULL$, for every vertex $x \neq u_0$;

$S \leftarrow \{u_0\}; i \leftarrow 0.$

Step 2: Computation

If $S = V$ goto Step 4;

Else, for each $x \in V - S$ do:

(i) If $l(x) > l(u_i) + \mathcal{W}(u_i, x)$, replace $l(x)$ by $l(u_i) + \mathcal{W}(u_i, x)$; $p(x) \leftarrow u_i$.

Else, retain $l(x)$ and $p(x)$.

(ii) Compute $\min\{l(x) : x \in V - S\}$.

(iii) Designate any vertex for which minimum is attained in (ii) as u_{i+1}.

(If there is more than one vertex for which the minimum is attained, you can designate any such vertex as u_{i+1}.)

Step 3: Recursion

$S \leftarrow S \cup \{u_{i+1}\}$; $i \leftarrow i + 1$; goto step 2.

Step 4: Output $l(x)$ as $d(u_0, x)$, and the array $P(x, u_0) = (x, p(x), p^2(x), \ldots, u_0)$, for each $x \in V(G)$ and STOP.

In the following, we illustrate the algorithm by taking a weighted graph.

Initialization:

$Vertexx$	a	b	c	d	e
$l(x)$	0	∞	∞	∞	∞

$u_0 = a$, $p(a) = a$, $S = \{a\}$; $V - S = \{b, c, d, e\}$.

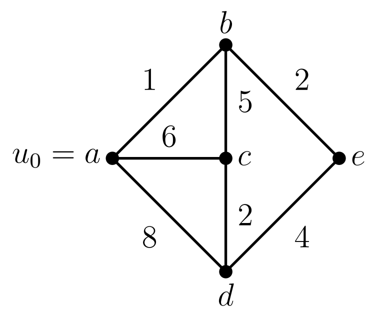


Figure 2.17: The input graph and its initialization.

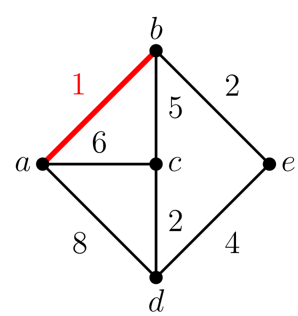
Iteration 1:

x	a	b	c	d	e
$l(x)$	0	1	6	8	∞

$\min\{l(x) : x \in V - S\} = 1$;

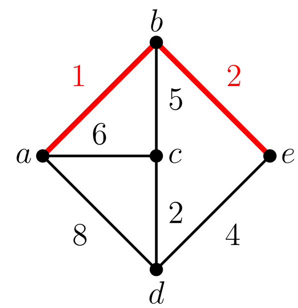
$l(b) = 1$; $p(b) = a$, $S = \{a, b\}$;

$V - S = \{c, d, e\}$, $P = (b, p(b)) = (b, a)$.



Iteration 2:

x	a	b	c	d	e
$l(x)$	0	1	6	8	3



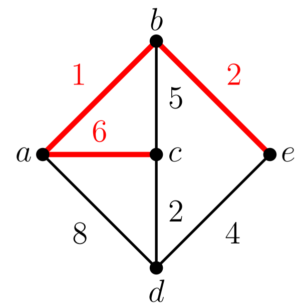
$\min\{l(x) : x \in V - S\} = 3;$

$l(e) = 3;\ p(e) = b,\ S = \{a, b, e\};$

$V - S = \{c, d\},\ P = (e, b, a).$

Iteration 3:

x	a	b	c	d	e
$l(x)$	0	1	6	7	3



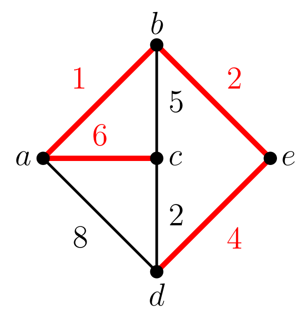
$\min\{l(x) : x \in V - S\} = 6;$

$l(c) = 6;\ p(c) = b,\ S = \{a, b, e, c\};$

$V - S = \{d\},\ P = (c, a).$

Iteration 4:

x	a	b	c	d	e
$l(x)$	0	1	6	7	3



$\min\{l(x) : x \in V - S\} = 7;$

$l(d) = 7;\ p(d) = e;\ S = \{a, b, e, c, d\};$

$V - S = \emptyset,\ P = (d, e, b, a).$

We stop the algorithm and declare the labels $l(x)$ shown in the above table as $d(a, x)$.

- The shortest (a, b)-path is (a, b).
- A shortest (a, c)-path is (a, c). The reader may notice that (a, b, c) is also a shortest (a, c)-path.
- The shortest (a, d)-path is (a, b, e, d).
- The shortest (a, e)-path is (a, b, e).

• Floyd-Warshall shortest path algorithm (1962)

It is a "all-to-all" algorithm in the sense that given a weighted graph (G, $\mathcal{W}$), the algorithm outputs the shortest distance $d(x, y)$, for every pair (x, y) of vertices. The algorithm makes use of a recursion formula proved below.

Theorem 2.24 ((Floyd and Warshall, 1962)). *Let (G, $\mathcal{W}$) be a weighted graph on vertices labeled* $1, 2, \ldots, n$. *Define*

$$d_{ij}^k = \begin{cases} \textit{The weighted length of a shortest } (i,j)\textit{-path with all its internal vertices} \\ \textit{from } \{1,\ldots,k\}\textit{; this path need not contain every vertex from } \{1,\ldots,k\}. \end{cases}$$

Then,

$$d_{ij}^k = \min\{d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}\}.$$

Proof. Let P be a shortest (i, j)-path with all its internal vertices from $\{1, \ldots, k\}$; so its length $l(P) = d_{ij}^k$. Two cases arise.

Case 1: k is not an internal vertex of P.

So P is a (i, j)-path with all its internal vertices from $\{1, 2, ..., k-1\}$. Hence, by defintion $d_{ij}^k = l(P) = d_{ij}^{k-1}$.

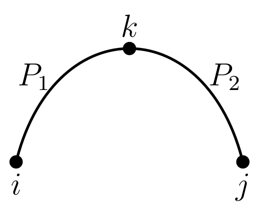


Figure 2.18: k is an internal vertex of P(i, j).

Case 2: k is an internal vertex of P.

In this case, P consists of two subpaths $P_1(i, k)$ and $P_2(k, j)$, where P_1 is a shortest (i, k)-path with all its internal vertices from $\{1, 2, \ldots, k-1\}$ and P_2 is a shortest (k, j) path with all its internal vertices from $\{1, 2, \ldots, k-1\}$. So, $l(P_1) = d_{ik}^{k-1}$, $l(P_2) = d_{kj}^{k-1}$. Hence,

$$d_{ij}^k = l(P) = l(P_1) + l(P_2) = d_{ik}^{k-1} + d_{kj}^{k-1}.$$

The two cases imply that $l(P)$ is either d_{ij}^{k-1} or $d_{ik}^{k-1} + d_{kj}^{k-1}$, whichever is minimum. So, $d_{ij}^k = \min \{d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}\}$. $\square$

Remarks. By the definition of d_{ij}^k, it may be observed that

- $d_{ij}^0 :=$ The weighted length of a shortest (i, j)-path with no internal vertices; so the path is (i, j).
- $d_{ij}^n :=$ The weighted length of a shortest (i, j)-path in G.
- $d_{ii}^n := 0$.

The input for the algorithm is an $n \times n$ matrix $W(G) = [\mathcal{W}_{ij}]$, called the ***weighted matrix***, where

$$\mathcal{W}_{ij} = \begin{cases} 0, & \text{if } i = j, \\ \mathcal{W}(i, j), & \text{if } i \neq j \text{ and } i,\ j \text{ are adjacent}, \\ \infty, & \text{if } i \neq j \text{ and } i,\ j \text{ are non-adjacent}. \end{cases}$$

The output is the $n \times n$ matrix $[d_{ij}^n]$, whose (i, j)th entry is the length of a shortest (i, j)-path.

◦ **Description of Floyd-Warshall algorithm**

Input: A weighted graph $(G, \mathcal{W})$ on vertices 1, 2,..., n.

Step 1: Initial: $D^0 \leftarrow W(G)$

Step 2: Recursion:

for $k = 1$ to n do

for $i = 1$ to n do

for $j = 1$ to n do

$$d_{ij}^k \leftarrow \min\{d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}\}.$$

Step 3: Output $D^n = [d_{ij}^n]$.

Remarks.

- An advantage of the algorithm is that it requires very little knowledge of program coding.
- The algorithm can be modified to output a shortest (x, y)-path; see exercises.

We again take the graph shown in Fig. 2.17 and illustrate the algorithm with vertices labeled $a = 1$, $b = 2$, $c = 3$, $d = 4$ and $e = 5$. In the following, we show the sequence of matrices D^0, D^1, D^2, D^3, D^4, D^5 successively generated by the algorithm.

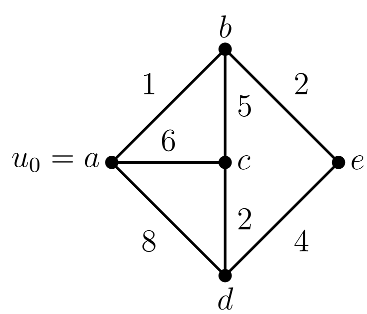


Figure 2.19: The input graph G.

Input matrix $\mathcal{W}(G) = D^0 = [d_{ij}^0]$.

	a	b	c	d	e
$a=1$	0	1	6	8	∞
$b=2$	1	0	5	∞	2
$c=3$	6	5	0	2	∞
$d=4$	8	∞	2	0	4
$e=5$	∞	2	∞	4	0

Output matrix $D^1 = [d^1_{ij}]$ **after one** (i,j)**-loop**, where $d^1_{ij} = \min\{d^0_{ij}, d^0_{i1} + d^0_{1j}\}$ indicates the length of a shortest (i,j)-path with internal vertices from $\{1 = a\}$.

	a	b	c	d	e
$a=1$	0	1	6	8	∞
$b=2$	1	0	5	9	2
$c=3$	6	5	0	2	∞
$d=4$	8	9	2	0	4
$e=5$	∞	2	∞	4	0

Output matrix after 2^{nd} (i,j) **loop**, where $d^2_{ij} = \min\ \{d^1_{ij},\ d^1_{i2}\ +\ d^1_{2j}\}$ indicates the length of a shortest (i,j)-path with internal vertices from $\{1 = a, 2 = b\}$.

	a	b	c	d	e
$a=1$	0	1	6	8	3
$b=2$	1	0	5	9	2
$c=3$	6	5	0	2	7
$d=4$	8	9	2	0	4
$e=5$	3	2	7	4	0

Output matrix after the 3^{rd} (i,j) **loop**, $D^3 = [d^3_{ij}]$, where $d^3_{ij} = \min\{d^2_{ij}, d^2_{i3} + d^2_{3j}\}$ indicates the length of a shortest (i,j)-path with internal vertices from $\{1 = a, 2 = b, 3 = c\}$.

	a	b	c	d	e
$a=1$	0	1	6	8	3
$b=2$	1	0	5	7	2
$c=3$	6	5	0	2	7
$d=4$	8	7	2	0	4
$e=5$	3	2	7	4	0

Output matrix after the 4^{th} (i,j) **loop**, $D^4 = [d^4_{ij}]$, where $d^4_{ij} = \min\{d^3_{ij}, d^3_{i4} + d^3_{4j}\}$ indicates the length of a shortest (i,j)-path with internal vertices from $\{1 = a, 2 = b, 3 = c, 4 = d\}$.

	a	b	c	d	e
$a=1$	0	1	6	8	3
$b=2$	1	0	5	7	2
$c=3$	6	5	0	2	6
$d=4$	8	7	2	0	4
$e=5$	3	2	6	4	0

Output matrix after the 5^{th} (i,j) **loop,** $D^5 = [d^5_{ij}]$, where $d^5_{ij} = \min\{d^4_{ij}, d^4_{i4} + d^4_{4j}\}$ indicates the length of a shortest (i,j)-path with internal vertices from $\{1 = a, 2 = b, 3 = c, 4 = d, 5 = e\}$ = The length of a shortest (i,j)-path.

	a	b	c	d	e
$a=1$	0	1	6	7	3
$b=2$	1	0	5	6	2
$c=3$	6	5	0	2	6
$d=4$	7	6	2	0	4
$e=5$	3	2	6	4	0

Exercises

1. Draw 3 mutually non-isomorphic graphs on 6 vertices and 5 edges, which do not contain cycles.

2. Give an example of a 3-regular graph on 10 vertices in which the minimum length of a cycle is 5.

3. Show that any two longest paths in a connected graph have a vertex in common. Does this assertion hold for a disconnected graph?

4. If C is a cycle in G, then an edge of G which joins two non-consecutive vertices of C is a called a ***chord*** of C. For example, $(2,5)$ is a chord of the 5-cycle $(1,2,3,4,5,1)$. Show that if G has an odd cycle then it has an odd cycle without chords.

5. Let G be a triangle-free graph and let C be a cycle of minimum length in G. Show that every vertex in $V(G) - V(C)$ is adjacent with at most two vertices of C.

6. If G is simple, connected, incomplete and $n \geq 3$, then show that G has three vertices u, v, w such that $(u,v), (v,w) \in E(G)$ but $(u,w) \notin E(G)$.

7. Give an example of a graph G such that
   (a) every two adjacent vertices lie on a common cycle,
   (b) there exists two adjacent edges that do not lie on a common cycle.

8. Let G be a graph. Define a relation R on $V(G)$ as follows. If $u, v \in V(G)$, then $u\ R\ v$ iff there exists a path connecting u and v. Show that R is an equivalence relation on G. What are the induced subgraphs $[V_1], [V_2], \ldots, [V_p]$, where $V_1, V_2, \ldots, V_p$ are the equivalence classes?

9. If G is simple and $\delta(G) \geq \frac{n-1}{2}$, then show that G is connected. Give an example of a disconnected simple graph G on 8 vertices with $\delta(G) = 3$.

10. If $m > \binom{n-1}{2}$ and G is simple, then show that G is connected. For each $n \geq 1$, find a disconnected graph with $m = \binom{n-1}{2}$.

11. If $deg(u) + deg(v) \geq p - 1$, for every pair of non-adjacent vertices in a simple graph G , then show that (i) G is connected, (ii) $diam(G) \leq 2$ and (iii) G has no cut-vertices.

12. Let G be a simple graph on vertices $v_1, v_2, \ldots, v_n$. Show that

(i) If every $G - v_i$ is disconnected, then G is also disconnected.

(ii) If $n \geq 3$ and $G - v_i$, $G - v_j$ are connected, for some $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$, then G is connected. Does this hold if $n = 2$?

13. Show the following for a connected graph G:
    (i) $c(G - v) \leq deg(v)$, for every $v \in V(G)$.
    (ii) If every vertex in G is of even degree, then $c(G - v) \leq \frac{1}{2}deg(v)$, for every $v \in V(G)$.

14. Let G ba a 2-connected simple graph. Let H be the simple graph obtained from G by adding a new vertex y and joining it with at least 2 vertices of G. Show that H is 2 connected.

15. Show that if G contains a closed walk of odd length, then it contains a cycle.

16. Show that G is connected iff every entry in $I + A + A^2 + \cdots + A^{n-1}$ is non-zero.

17. Find the maximum number of edges in a simple graph G which has girth 6.

18. Show that a k-regular simple graph with girth four has at least $2k$ vertices; draw such a graph on $2k$ vertices.

19. Show that a k-regular simple graph of girth five has at least $k^2 + 1$ vertices. In addition, if G has diameter two, then show that it has exactly $k^2 + 1$ vertices. Draw such a graph on $2k$ vertices.

20. Prove or disprove: If G is a connected graph with cut-vertices and if u and v are vertices of G such that $d(u, v) = diam(G)$, then no block of G contains both u and v.

21. Show that every graph with a cut-vertex has at least two end-blocks. (A block H of G is called an **end-block** if it contains exactly one cut-vertex of G.)

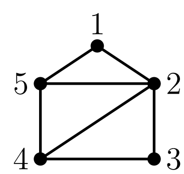
22. Draw the following simple graphs:
    (a) A graph on 10 vertices with 3 components that has maximum number of edges.
    (b) A connected graph (on at most 10 vertices) which has exactly 3 cut-vertices and exactly 2 cut-edges.

23. Show that every k-connected graph G has at least $\frac{nk}{2}$ edges.

24. If G is a 2-connected graph, then show that for some $(x, y) \in E(G)$, $G - x - y$ is connected.

25. If the average degree of a connected graph G is greater than two, prove that G has at least two cycles.

26. Draw connected spanning subgraphs H_1, H_2 and H_3 of the graph G (shown below) having diameters 2,3 and 4 respectively and with minimum number of edges.



G

27. Give an example of a graph with the following properties or explain why no such example exists.
    (i) A 4-connected graph that is not 3-connected.
    (ii) A 3-connected graph that is not 4-connected.
    (iii) A 2-edge-connected graph that is not 3-edge-connected.
    (iv) A 3-edge-connected graph that is not 2-edge-connected.

28. (a) If G is disconnected, then show that G^c is connected. Is the converse true?
    (b) If G is simple, show that G or G^c has diameter ≤ 3.
    (c) If G is self-complimentary, show that $2 \leq diam(G) \leq 3$.

29. For any two graphs G_1 and G_2, show the following.
    (a) $k_0(G_1 \cup G_2) = 0$.
    (b) $k_0(G_1 + G_2) = \min\{k_0(G_1) + n(G_2), k_0(G_2) + n(G_1)\}$.
    (c) $k_1(G_1 \cup G_2) = 0$.
    (d) $k_1(G_1 + G_2) = \delta(G_1 + G_2)$.

30. Give an example of a graph G with the following properties or state why such examples do not exist.
    (a) $k_0(G) = 3$, $k_1(G) = 4$ and $\delta(G) = 5$

(b) $k_0(G) = 2$, $k_1(G) = 2$ and $\delta(G) = 4$

(c) $k_0(G) = 3$, $k_1(G) = 3$ and $\delta(G) = 2$

(d) $k_0(G) = 3$, $k_1(G) = 2$ and $\delta(G) = 4$

31. Prove or disprove:
   (i) If G is a graph with $k_0(G) = k \geq 1$, then $G - U$ is disconnected, for every set U of k vertices.
   (ii) If G is a connected graph and U is a minimum vertex-cut, then $G - U$ contains exactly two components.
   (iii) If G is a graph on n vertices and $W = \{u \in V(G) : deg(u) = n - 1\}$, then
      (a) $W \subseteq K(G)$, for every vertex-cut $K(G)$ of G.
      (b) every edge-cut of G contains an edge incident with a vertex in W.

32. Find the connectivity and edge-connectivity of the d-cube Q_d.

33. Prove that if G is a simple r-regular graph, where $0 \leq r \leq 3$, then $k_0(G) = k_1(G)$. Does it hold for multi-graphs? Find the minimum positive integer r for which there exists an r-regular graph G such that $k_0(G) \neq k_1(G)$.

34. Find the minimum positive integer r for which there exists a r-regular graph such that $k_1(G) \geq k_0(G) + 2$.

35. If G is simple and $\delta(G) \geq n - 2$, then show that $k_0(G) = \delta(G)$.

36. If G is simple and $\delta(G) \geq \frac{n}{2}$, then show that $k_1(G) = \delta(G)$.

37. If G is a simple graph with $\delta(G) \geq \frac{n+1}{2}$, then show that G is 3-vertex connected.

38. If G is simple and has no even cycles, then show that each block of G is either K_2 or an odd cycle.

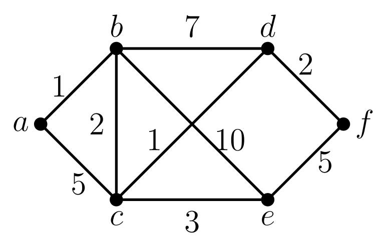
39. Draw a network N with 7 nodes and with minimum number of links satisfying the following conditions. Justify that your network has minimum number of links.
   (i) N contains no self-loops and multiple links.
   (ii) If any two nodes fail, the remaining 5 healthy nodes can communicate along themselves.

40. Let G be an incomplete simple graph on n vertices with vertex connectivity k. If $deg(v) \geq \frac{1}{3}(n + 2k - 2)$, for every vertex v, and S is a vertex-cut with k vertices, then find the number of components in $G - S$.

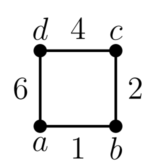
41. If G is a connected graph, then G^2 has vertex set V(G) and two vertices u, v are adjacent in G^2 iff the distance between u and v in G is 1 or 2.
    (a) Draw $(P_6)^2$, where P_6 is the path on 6 vertices.
    (b) If G is connected, show that G^2 is 2-connected.

42. Let G be a simple graph with exactly one cycle. Find all possible values of the vertex connectivity and edge connectivity of G. Justify your answer.

43. Use Dijkstra's algorithm to compute a shortest path from the vertex a to every other vertex in the edge weighted graph shown below. Show the weighted paths generated at each step of the algorithm.



44. Find the entry in the first row and second column of the matrix generated after applying one iteration of the Floyd-Warshall algorithm to the weighted graph shown below.



45. If $k_0(G) \geq 3$, then show that $k_0(G - e) \geq 2$, for every edge e.

Module 3 Trees

Contents

Trees are omnipresent in computer science and network analysis. The earliest mention of trees appear in the works of Kirchoff (1847), Jordan (1869), Sylvester (1882) and Cayley (1957).

3.1 Definitions and characterizations

Definitions.

- A connected acyclic graph is called a ***tree***.
- An acyclic graph is called a ***forest***.

  Thus, every component of a forest is a tree.

All the non-isomorphic trees up to seven vertices are shown below.

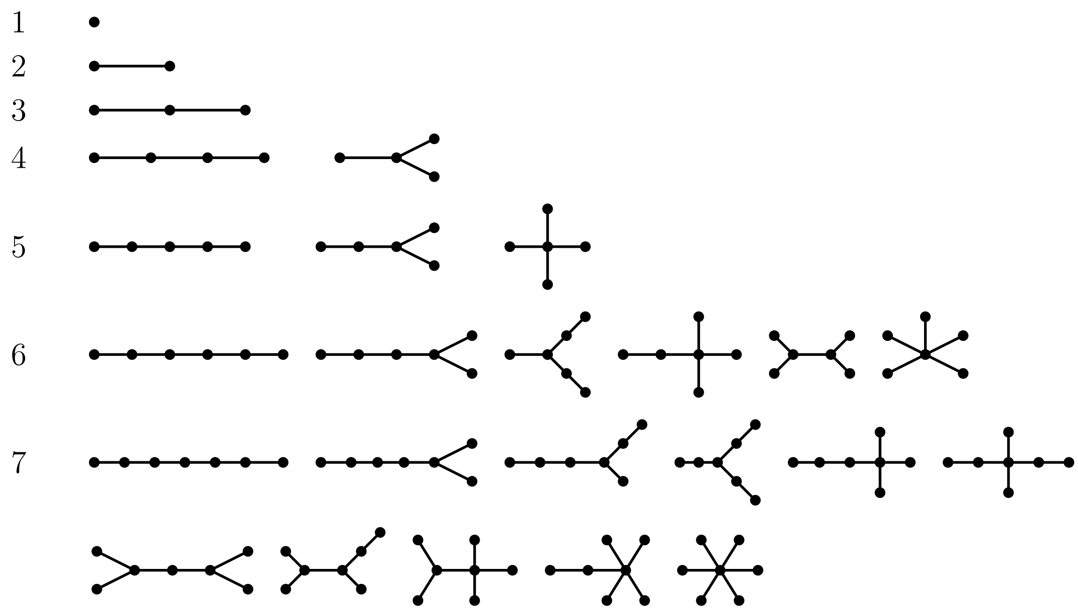


Figure 3.1: Non-isomorphic trees up to seven vertices.

Because of their simple structure, trees can be characterized in many ways. Some are proved below and some are included in the exercise list.

Theorem 3.1. *For any graph G, the following statements are equivalent.*

(a) G is a tree.

(b) G is connected and $m = n - 1$.

(c) G is acyclic and $m = n - 1$.

Proof. $(a) \Rightarrow (b)$: Since G is a tree, it is connected and therefore $m \geq n - 1$, by Corollary to Theorem 2.6. If $m \geq n$, then G contains a cycle, by Theorem 2.3, contrary to our hypothesis. Hence $m = n - 1$.

$(b) \Rightarrow (c)$: We have to only show that G is acyclic. On the contrary, assume that G contains a cycle. If e is an edge of this cycle, then $G - e$ is connected (by Theorem 2.15), and it has n vertices and $n - 2$ edges, a contradiction to Corollary of Theorem 2.6.

$(c) \Rightarrow (a)$: Since $m = n - 1$, G is connected or contains a cycle by Theorem 2.7. But by our hypothesis, G is acyclic. Therefore, G is connected and acyclic, and so it is a tree. □

Theorem 3.2. *A graph G is a tree if and only if any two vertices are connected by a unique path.*

Proof. (1) G is a tree $\Rightarrow$ Any two vertices are connected by a unique path.

Any two vertices are connected by a path, since G is connected. Next suppose that there exist two vertices u and v which are connected by two paths $P(u, v)$ and $Q(u, v)$. They may be internally disjoint or they may have a common vertex. In the latter case, let $z \neq v$ be the last common vertex. (It is possible that $z = u$.) Then $P(z, v)$ and $Q(z, v)$ are two internally disjoint (z, v)-paths which together form a cycle in G, a contradiction to the acyclicity of G.

(2) Any two vertices are connected by a unique path $\Rightarrow$ G is a tree.

Clearly, G is connected. We next claim that G is acyclic. Suppose G contains a cycle C. If we choose any two vertices u, v on this cycle, they are connected by two paths, a contradiction to our hypothesis. □

Theorem 3.3. *A connected graph G is a tree if and only if every edge is a cut-edge.*

Proof. It is a consequence of Theorems 3.2 and 2.15. □

Theorem 3.4. *Every connected graph G contains a subgraph which is a spanning tree.*

Proof. By successively deleting an edge of a cycle as long as there are cycles, we get a spanning tree of G. □

Theorem 3.5. *Every tree T on n vertices $(n \geq 2)$ contains at least two vertices of degree 1.*

Proof. Let p be the number of vertices of degree 1 in T. Then

$$2(n-1) = 2m = \sum_{v \in V(T)} deg(v) \geq 2(n-p) + p.$$

Hence, $p \geq 2$. □

Previous two theorems imply that every connected graph G on n vertices contains a tree on k vertices for every k, $1 \leq k \leq n$. However, G need not contain *every* tree on k vertices, for a given k, $1 \leq k \leq n$. See Figure 3.2.

The following result gives a sufficient condition for a graph G to contain every tree on a given number of vertices.

Theorem 3.6. *(Chvatal, 1977) Let T be any tree on $k+1$ vertices. If $\delta(G) \geq k$, then G contains a tree (isomorphic to) T.*

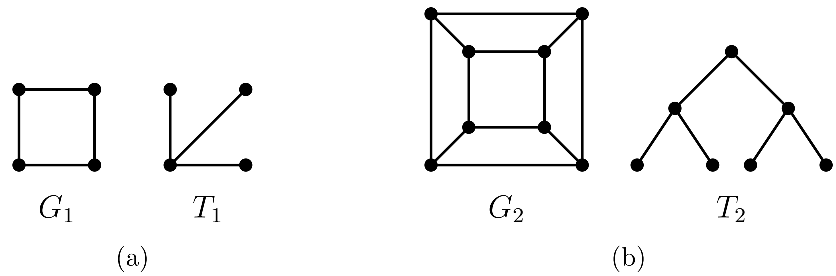


Figure 3.2: $T_1 \not\subseteq G_1$ and $T_2 \not\subseteq G_2$.

Proof. We prove the result by induction on k.

Basic step: If $k = 0$, the result is obvious.

Induction step ($k \geq 1$): Assume that if T' is any tree on k vertices and $\delta(G) \geq k-1$, then G contains T'. We show that if T is any tree on $k+1$ vertices and $\delta(G) \geq k$, then G contains T. By Theorem 3.5, T contains a vertex z of degree 1. Let y be the unique vertex adjacent with z. $T - z$ is a tree on k vertices. By induction hypothesis,

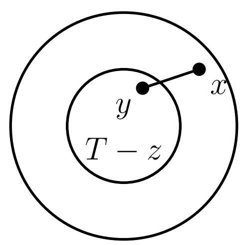


Figure 3.3: Construction of T from $T - z$.

G contains $T - z$. Since $deg_G(y) \geq \delta(G) \geq k + 1$ and $T - z$ contains k vertices, we conclude that y is adjacent with a vertex x in $V(G) - V(T - z)$; see Figure 3.3. Then the tree $T - z + (x, y)$ is isomorphic with T. $\square$

3.2 Number of trees (Optional)

In this section, we are concerned with the following problem.

How many trees are there on a given set of n vertices, where two trees T_1 and T_2 are counted distinct if $E(T_1) \neq E(T_2)$?

For simplicity, we assume that the vertices are $1, 2, \ldots, n$. And denote the number of trees on n vertices by $\mathcal{J}_n$. All the trees on $n = 1$ to 4 are shown below. Therefore $\mathcal{J}_1 = 1$, $\mathcal{J}_2 = 1$, $\mathcal{J}_3 = 3$, $\mathcal{J}_4 = 16$.

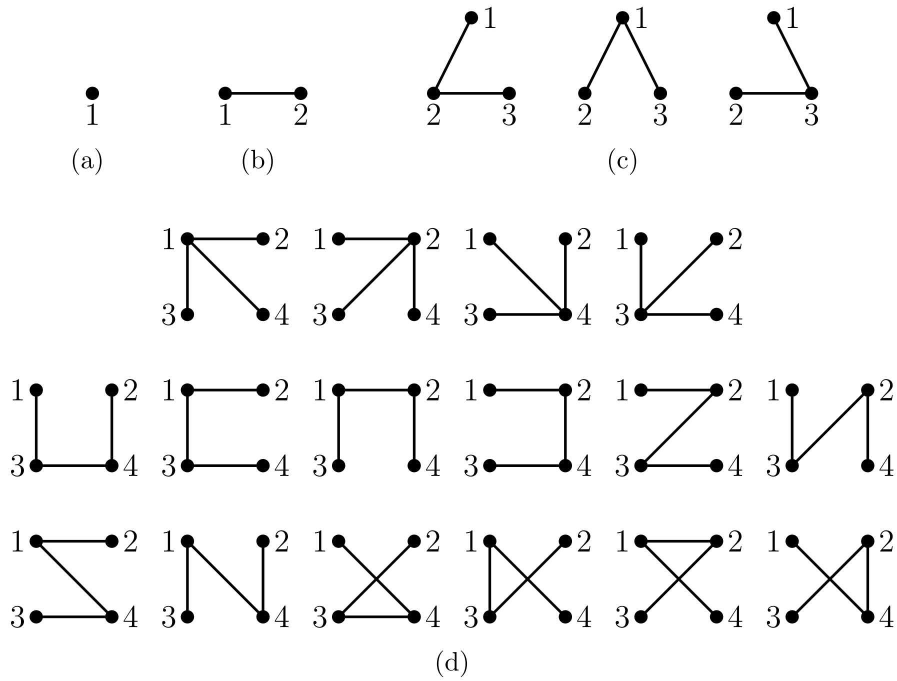


Figure 3.4: Distinct trees upto four vertices.

It is important to note that $\mathcal{J}_n$ is not the number of non-isomorphic trees on n vertices. For example, $\mathcal{J}_3 = 3$, whereas there is only one non-isomorphic tree on three vertices; and $\mathcal{J}_4 = 16$, whereas there are only two non-isomorphic trees on four vertices. In Figure 3.4d, first four trees are mutually isomorphic and next twelve trees are mutually isomorphic.

• Cayley's formula

In 1889, Arthur Cayley proved a formula for the number of distinct trees on n given vertices. This he did towards his research on counting chemical isomers C_nH_{2n+2}. Since then there have been at least 20 new proofs. We give a proof due to Prüfer. This proof technique is widely used in discrete mathematics. Suppose we would like to count the number of objects in a given set S. Instead of counting the objects in S, we find a set T whose cardinality we know. We then establish a bijection from S to T which shows that $|S| = |T|$. Instead of establishing a bijection, we may establish an injection from S to T and an injection from T to S and appeal to Shroeder-Bernstein theorem and conclude that $|S| = |T|$.

Theorem 3.7 (Cayley's formula for the number of distinct trees)**.** *The number of distinct trees on n vertices is n^{n-2}.*

Proof. (Prüfer, 1918) It is an indirect proof. Let $S_n = \{(t_1, t_2, \ldots, t_{n-2}) : t_i \in \{1, 2, \ldots, n\}\}$. Clearly, $|S_n| = n^{n-2}$. Let F_n be the set of all distinct trees on n vertices; so $|F_n| = \mathcal{J}_n$. We first show that there is a one-one map from F_n to S_n, and next show that there is one-one map from S_n to F_n. It follows that there exists a one-one and onto map from F_n to S_n, and so $\mathcal{J}_n = |F_n| = |S_n| = n^{n-2}$.

(1) One-one map from F_n to S_n:

With a given tree $T \in F_n$, we associate a unique $(n-2)$-tuple $t = (t_1, t_2, \ldots, t_{n-2})$ by defining $t_1, t_2, \ldots, t_{n-2}$.

Definition of t_1: Among all the vertices of degree one in T, let s_1 be the vertex with the least label. We designate the unique vertex that is adjacent with s_1 as t_1.

Definition of t_2: Consider the tree $T - s_1$. Among all the vertices of degree one in $T - s_1$, let s_2 be the vertex with the least label. We designate the unique vertex that is adjacent with s_2 as t_2.

Definition of t_3*:* Consider the tree $T - s_1 - s_2$. Among all the vertices of degree one in $T - s_1 - s_2$, let s_3 be the vertex with the least label. We designate the unique vertex adjacent with s_3 as t_3.

This process is continued until t_{n-2} is defined.

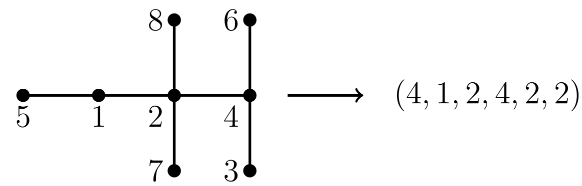


Figure 3.5: A tree T and its Prüfer code.

This association of t with T is a one-one map since t is unique, and moreover if $T = T'$, then $E(T) = E(T')$ and so $t = t'$.

- t is called the ***Prüfer code*** of T.

(2) One-one map from S_n to F_n:

Given a $(n-2)$-tuple $t = (t_1, t_2, \ldots, t_{n-2}) \in S_n$, we construct a unique tree $T \in F_n$. We start with a forest F of n isolated vertices $1, 2, \ldots, n$ and proceed to add $n - 1$ edges, one at a time, to obtain T. Let $N = \{1, 2, \ldots, n\}$.

Addition of first edge: Let $s_1 \in N$ be the least element which does not appear in $(t_1, t_2, \ldots, t_{n-2})$. Join s_1 and t_1.

Addition of second edge: Consider $N - \{s_1\}$ and $(t_2, t_3, \ldots, t_{n-2})$. Let $s_2 \in N - \{s_1\}$ be the least element which does not appear in $(t_2, \ldots, t_{n-2})$. Join s_2 and t_2.

Addition of third edge: Consider $N - \{s_1, s_2\}$ and $(t_3, t_4, \ldots, t_{n-2})$. Let $s_3 \in N - \{s_1, s_2\}$ be the least element which does not appear in $(t_3, t_4, \ldots, t_{n-2})$. Join s_3 and t_3.

This process is continued until $s_{n-2} \in N - \{s_1, s_2, \ldots, s_{n-3}\}$ is identified and joined to t_{n-2}. Until now we have added $n-2$ edges to F and we are left with two elements $s_{n-1}, s_n \in N - \{s_1, s_2, \ldots, s_{n-2}\}$. Finally, join s_{n-1} and s_n. In Figure 3.6, we have illustrated this construction by taking the 6-tuple (4, 1, 2, 4, 2, 2) and get back the tree of Figure 3.5. □

• Kirchoff-matrix-tree theorem

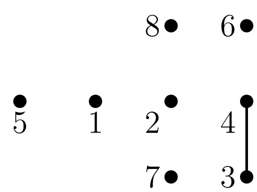
Kirchoff (1847) developed the theory of trees to find the current in each branch and around each circuit in an electrical network; any text book on network analysis contains these details. In this context, he was interested in finding the spanning trees of a given graph (representing an electrical network). Figure 3.7 shows 8 distinct spanning trees of a graph.

Remarks.

(1) The number of spanning trees of a graph G is denoted by $\mathcal{J}(G)$, where two spanning trees T_1 and T_2 are counted as distinct if $E(T_1) \neq E(T_2)$.

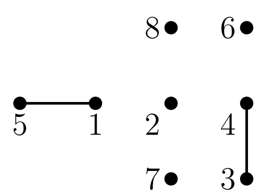
(2) $\mathcal{J}(G) = 0$ if and only if G is disconnected.

$\{1, 2, 3, 4, 5, 6, 7, 8\}$
$(4, 1, 2, 4, 2, 2)$
$s_1 = 3,\ t_1 = 4.$



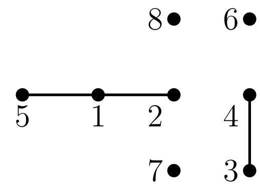
(a)

$\{1, 2, 4, 5, 6, 7, 8\}$
$(1, 2, 4, 2, 2)$
$s_2 = 5,\ t_2 = 1.$



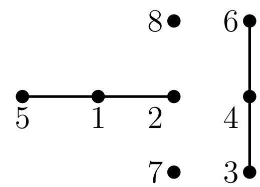
(b)

$\{1,2,4,6,7,8\}$
$(2,4,2,2)$
$s_3 = 1,\ t_3 = 2.$



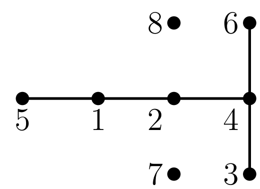
(c)

$\{2,4,6,7,8\}$
$(4,2,2)$
$s_4 = 6,\ t_4 = 4.$



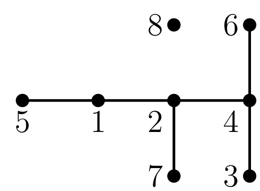
(d)

$\{2,4,7,8\}$
$(2,2)$
$s_5 = 4,\ t_5 = 2.$



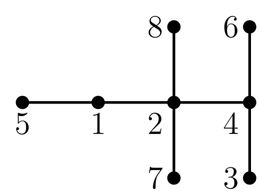
(e)

$\{2,7,8\}$
(2)
$s_6 = 7,\ t_6 = 2.$



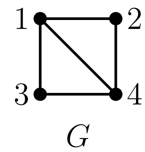
(f)

$\{2,8\}$



(g)

Figure 3.6: Construction of a tree with a given Prüfer code (4,1,2,4,2,2).



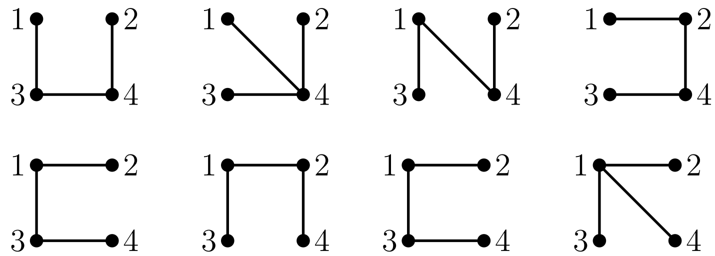


Figure 3.7: G and its eight distinct spanning trees

(3) $\mathcal{J}(G) = 1$ if and only if G is a tree.

(4) $\mathcal{J}(K_n) = \mathcal{J}_n = n^{n-2}$ (Cayley's formula).

(5) If $G^{(k)}$ denotes the graph obtained by replacing every edge of a simple graph G with k parallel edges, then $\mathcal{J}(G^{(k)}) = k^{n-1}\mathcal{J}(G)$.

(6) If G_0 denotes the simple graph obtained from a graph G by deleting all the loops of G, then $\mathcal{J}(G_0) = \mathcal{J}(G)$. So, in finding $\mathcal{J}(G)$ one can ignore the loops.

Kirchoff discovered an amazing formula for $\mathcal{J}(G)$. Cayley's formula can be derived using Kirchoff's formula.

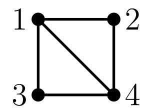
Theorem 3.8 (Kirchoff-Matrix-Tree-Theorem)**.** *Let G be a loopless graph and $A(G)$ be its adjacency matrix. Let $K(G)$ denote the $n \times n$ matrix obtained from $A(G)$ by applying the following two operations:*

1. *Change the sign of every off-diagonal entry from $+$ to $-$.*
2. *Replace the i-th diagonal entry by the degree of the i-th vertex.*

Then all the $(n-1)\times(n-1)$-co-factors of $K(G)$ have the same value and the common value is $\mathcal{J}(G)$, that is $\mathcal{J}(G) = (-1)^{s+t}det(M_{(s,t)})$, where $M_{(s,t)}$ is the $(n-1)\times(n-1)$ matrix obtained from $K(G)$ by deleting the s-th row and the t-th column. □

- $K(G)$ is called the ***Kirchoff matrix***.

Illustration 1:



(a) G

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

(b) $A(G)$

$$\begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}$$

(c) $K(G)$

$$\begin{bmatrix} 3 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 3 \end{bmatrix}$$

(d) $M(2,3)$

Figure 3.8: A graph G, its adjacency matrix, associated Kirchoff matrix and a co-factor.

$\mathcal{J}(G) = (-1)^{2+3} det(M(2,3)) = 8$. The eight distinct spanning trees of G are shown in Figure 3.7.

Illustration 2: Derivation of Cayley's formula using Kirchoff-Matrix-Tree Theorem.

Apply row/column operations to $M(1,1)$ and reduce it to the following triangular matrix (for example, $R_1 \leftarrow R_1 + R_2 + \cdots + R_{n-1}$, $R_i \leftarrow R_1 + R_i$, for $2 \leq i \leq n-1$).

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & n & 0 & \dots & 0 \\ 0 & 0 & n & \dots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \dots & n \end{bmatrix}_{(n-1)\times(n-1)}$$

$\mathcal{J}(K_n) = (-1)^2 det(M(1,1)) = n^{n-2}$.

$$\begin{bmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 1 & 0 & \dots & 1 \\ & & \vdots & & \\ 1 & 1 & 1 & \dots & 0 \end{bmatrix}_{n\times n}$$

(a) $A(K_n)$

$$\begin{bmatrix} n-1 & -1 & -1 & \dots & -1 \\ -1 & n-1 & -1 & \dots & -1 \\ -1 & -1 & n-1 & \dots & -1 \\ & & \vdots & & \\ -1 & -1 & -1 & \dots & n-1 \end{bmatrix}_{n\times n}$$

(b) $K(K_n)$

$$\begin{bmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ & & \vdots & \\ -1 & -1 & \dots & n-1 \end{bmatrix}_{(n-1)\times(n-1)}$$

(c) $M(1,1)$

3.3 Minimum spanning trees

A real world problem whose algorithmic solution requires huge amount of time (technically, super exponential time) by brute force method but admits a decent (technically, low polynomial time) algorithm is the following. In fact, the algorithm follows "greedy paradigm". A few towns are to be connected by a network of roads. The cost of road construction between any two towns is given. The problem is to design a road network such that (i) given any two towns one is reachable from the other; and (ii) the total cost of the network is minimum possible.

It is clear that the problem can be easily modeled as a graph theoretic problem: Represent the towns by vertices and join two vertices by an edge e of weight $\mathcal{W}(e)$, if the cost of the road construction between the two corresponding towns is $\mathcal{W}(e)$. The problem is then to find a connected spanning graph H of a weighted graph G such that $\sum_{e\in H} \mathcal{W}(e)$ is minimum among all connected subgraphs H of G. We may assume that the weights are positive as they represent costs. It is immediate that such a H must be a tree; if H contains a cycle, then by deleting an edge of this cycle we get a connected subgraph H' of H whose weight is obviously less than that of H.

Definition. *A spanning tree of minimum weight in a connected weighted graph G is called a **minimum spanning tree** of G.*

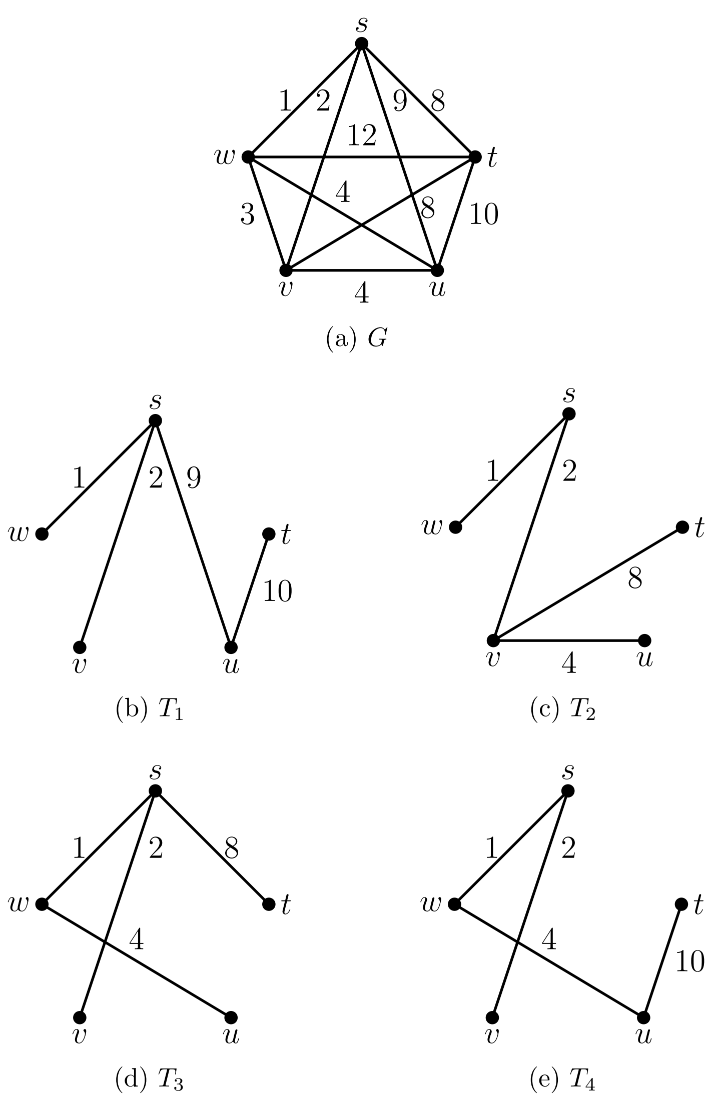


Figure 3.9: A weighted graph G and four of its spanning trees T_1, T_2,T_3, and T_4, with $\mathcal{W}(T_1) = 22$, $\mathcal{W}(T_2) = 15$, $\mathcal{W}(T_3) = 15$ and $\mathcal{W}(T_4) = 17$.

In the following we describe two algorithms to find a MST of a connected weighted graph G.

• Kruskal's algorithm (J.B. Kruskal, 1956)

Input: A weighted connected graph (G, $\mathcal{W}$)

Output: A minimum spanning tree T_k of G.

Step 1 (Initial): Arrange the edges of G in non-decreasing order, say π. Define H_0 to be the graph with vertex set $V(G)$ and no edges. (During the execution of the algorithm H_0 is updated by the addition of edges to yield T_k.)

Step 2: Select the first edge from π say e_1 and add it to H_0. Call the resulting graph H_1.

Step 3 (Recursion): After selecting i edges $e_1, e_2, \ldots, e_i$ and forming the graph H_i, find the next edge, say e_{i+1}, in the subsequence succeeding e_i in π which does not create a cycle with the edges already selected. Add it to H_i and call the resulting graph H_{i+1}.

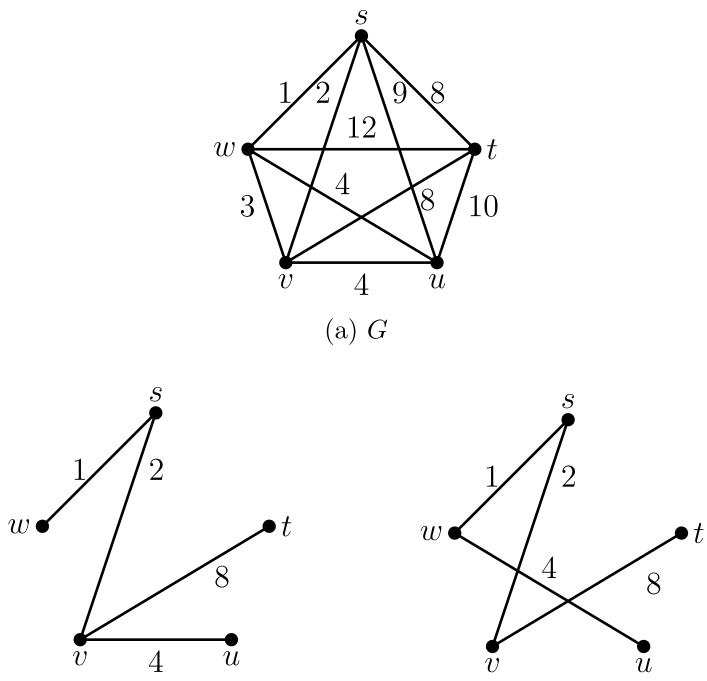
Step 4 (Stop Rule): Stop when $n-1$ edges are selected. Output H_{n-1} as T_k.

An illustration for Kruskal's algorithm

Consider the weighted graph shown in Figure 3.9. Arrange the edges in non-decreasing order with respect to their weight

$$((s,w),(s,v),(v,w),(v,u),(u,w),(s,t),(v,t),(s,u),(u,t),(t,w)).$$

As described in Kruskal's algorithm we successively select the edges (s,w), (s,v), (v,u) and (v,t), and obtain the first MST (MST_1) shown in Figure 3.10b. Note that at the third iteration, although the edge (v,w) is of minimum weight it cannot be selected as it creates a cycle with the two edges already selected.



(a) G

(b) MST_1

(c) MST_2

Figure 3.10: Two minimum spanning trees of G constructed by Kruskal's algorithm. Notice that both the trees have the same weight 15 but they are not isomorphic.

By choosing a different ordering of the edges $((s,w),(s,v),(v,w),(u,w),(v,u),$ $(v,t),(s,t),(s,u),(t,u),(t,w))$ and applying Kruskal's algorithm, we get the second minimum spanning tree shown in Figure 3.10c.

Theorem 3.9. *Any tree T_k constructed by Kruskal's algorithm is a MST of G.*

Proof. At the outset we observe that T_k is indeed a tree. This follows since H_{n-1} has $n-1$ edges and each H_i is acyclic (see step 3). Let T_{min} be a MST of G. If $E(T_k)$ $= E(T_{min})$, we are through. Next, suppose that $E(T_k) \neq E(T_{min})$ and let e_{i+1} be the first edge in T_k which is not in T_{min}. So, $E(T_k) = \{e_1, e_2, ..., e_i, e_{i+1}, ..., e_{n-1}\}$ and $E(T_{min}) = \{e_1, e_2, ..., e_i, f_{i+1}, ..., f_{n-1}\}$. Consider the graph $T_{min} + e_{i+1}$. It contains a unique cycle, say C; see Figure 3.11. Since $E(C) \nsubseteq E(T_k)$, there exists an edge $f \in E(C)$ which is in $E(T_{min}) - E(T_k)$.

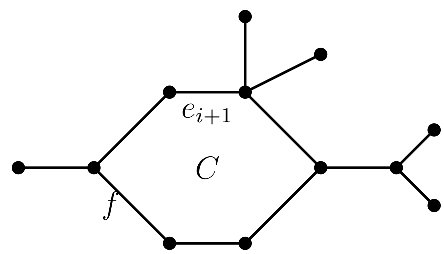


Figure 3.11: $T_{min} + e_{i+1}$.

Consider the tree $T_{min} + e_{i+1} - f$. We have $\mathcal{W}(T_{min}) \leq \mathcal{W}(T_{min} + e_{i+1} - f)$ $= \mathcal{W}(T_{min}) + \mathcal{W}(e_{i+1}) - \mathcal{W}(f)$. Therefore, $\mathcal{W}(f) \leq \mathcal{W}(e_{i+1})$. On the other hand, f too creates no cycles with the edges $e_1, e_2, \ldots, e_i$, since $e_1, e_2, \ldots, e_i, f \in T_{min}$. But we have selected e_{i+1} in the $(i+1)^{th}$ iteration. Therefore, $\mathcal{W}(e_{i+1}) \leq \mathcal{W}(f)$. Hence, $\mathcal{W}(e_{i+1}) = \mathcal{W}(f)$ and so $\mathcal{W}(T_{min}+e_{i+1}-f) = \mathcal{W}(T_{min})$. That is, $T_{min}+e_{i+1}-f$ is also a MST with one more edge in common with T_k. If $E(T_k) \neq E(T_{min} + e_{i+1} - f)$, we continue the above process enough number of times to conclude that T_k is a MST. $\square$

• Prim's algorithm (R. C. Prim, 1957)

Input: A weighted connected graph (G, $\mathcal{W}$)

Output: A minimum spanning tree T_p of G.

Idea: If we have a subtree T of G and if we take a new vertex $x \in V(G) - V(T)$, and join it to precisely one vertex of T, then the resultant graph $T + x$ is also a tree.

Step 1: Select a vertex v_1 (arbitrarily). Select an edge of minimum weight in $[\{v_1\}, V - \{v_1\}]$, say (v_1, v_2). Define H_1 to be the tree with vertices v_1, v_2 and the edge (v_1, v_2).

Step 2: Having selected the vertices, $v_1, v_2, \ldots, v_k$ and $k-1$ edges, and forming the tree H_{k-1}, select an edge of minimum weight in $[\{v_1, \dots, v_k\}, V - \{v_1, \dots, v_k\}]$, say (v_i, v_{k+1}). Define H_k to be the tree obtained by joining v_{k+1} to $v_i \in H_{k-1}$.

Step 3 (Termination): Stop when $n-1$ edges are selected and output H_{n-1} as T_p.

An illustration:

Again consider the weighted K_5 shown in Figure 3.9. The successive selection of edges (v, s), (s, w), (u, w), and (t, v), yields the MST shown below.

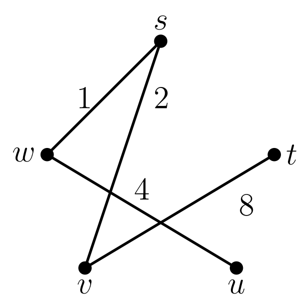


Figure 3.12: MST constructed using Prim's algorithm

Theorem 3.10. *Any tree T_p constructed by Prim's algorithm is a MST.*

Proof. As remarked in the description of the algorithm, T_p is indeed a tree. Let $e_1, e_2, \dots, e_n$ be the order of edges in which they were selected to construct T_p. To prove the theorem it suffices to prove the following assertion:

(*) If there is a MST T with $e_1, e_2, \dots, e_{k-1} \in T$ and $e_k \notin T$, then there is a MST T' with $e_1, e_2, \dots, e_{k-1}, e_k \in T'$.

So, let T be a MST with $e_1, e_2, \dots, e_{k-1} \in T$ and $e_k \notin T$. Since $e_k = (v_i, v_{k+1})$ and $v_i \in \{v_1, \dots, v_k\}$, we have $v_{k+1} \in V - \{v_1, \dots, v_k\}$; see Step 2. $T + e_k$ contains a unique cycle. So there exists an edge $f = (u, v)$, where $u \in \{v_1, v_2, \dots, v_k\}$ and $v \in \{v_{k+1}, \dots, v_p\}$; see Figure 3.13. Consider the tree $T' = T + e_k - f$. Since at the k-th stage we have preferred to select e_k, it follows that $\mathcal{W}(e_k) \leq \mathcal{W}(f)$. Hence

$\mathcal{W}(T + e_k - f) = \mathcal{W}(T) + \mathcal{W}(e_k) - \mathcal{W}(f) \leq \mathcal{W}(T)$. Since T is a MST, we have $\mathcal{W}(T+e_k-f) = \mathcal{W}(T)$. So $T+e_k-f$ is also a MST of G containing $e_1, e_2, \ldots, e_{k-1}, e_k$. Thus (*), and hence the theorem is proved.

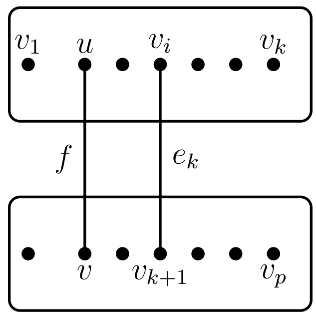


Figure 3.13: Construction of T' from $T + e_k$.

$\square$

Exercises

1. Draw two non-isomorphic trees with the same degree sequence. What is the smallest n for the existence of such trees?

2. Suppose G is a simple connected graph such that $G - v$ is acyclic, for every vertex $v \in V(G)$. Determine the number of edges in G, and determine G itself.

3. If G is a tree with all the degrees odd, then show that the number of edges in G is odd.

4. Find a tree realization and a disconnected graph realization of the sequence $(1^6, 2, 3^2, 4)$.

5. (a) If a tree has degree sequence $(1^k, 4^p)$, then find k (in terms of p).

   (b) Which of the quizzes stated on the beginning of this course is now solved.

6. Let $n \geq 2$. Show that a sequence $(d_1, d_2, \ldots, d_n)$ of integers is the degree sequence of a tree iff (i) $d_i \geq 1$, for $1 \leq i \leq n$ and (ii) $\sum_{i=1}^{n} d_i = 2n - 2$.

7. Let G be a forest with k components. How many edges should one add to G, so that the resulting graph is a tree?

8. Show that there exists a connected graph with degree sequence $d_1 \geq d_2 \geq \ldots \geq d_n$ iff (i) $d_i \geq 1$, for $1 \leq i \leq n$ and (ii) $\sum_{i=1}^{n} d_i \geq 2n - 2$.

9. Show that in a tree T there are at lest $\Delta(T)$ vertices of degree 1.

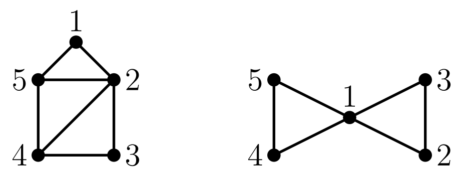
10. If G is a simple, 3-regular, connected graph, then show that in every spanning tree T of G, $n_1(T) = n_3(T) + 2$, where $n_i(T) =$ the number of vertices of degree i in T.

11. A tree T on 10 vertices has exactly one vertex of degree 4 and exactly one vertex of degree 3. Find the maximum degree of T and find all the possible degree sequence of T. Draw three such non-isomorphic trees.

12. The maximum degree of a tree T is 5 and let $n_i(T)$ denote the number of vertices of degree i. If $n_1(T) = 50$ and $n_2(T) = n_3(T) = n_4(T) = n_5(T)$, find the number of vertices in T.

13. Let $P_n^{(2)}$ be the graph obtained by doubling every edge of P_n. Find the number of spanning trees of $P_n^{(2)}$. For example: $P_3^{(2)} :=$ 

14. Find $\mathcal{J}(G_1)$ and $\mathcal{J}(G_2)$, where G_1 and G_2 are shown below, using Kirchoff-matrix-tree theorem.



(a) G_1 (b) G_2

Trees are extensively used in data structures. These trees in fact have more structure. A vertex v of a tree T, chosen arbitrarily and distinguished from others, is called the ***root of*** T; T is called a ***rooted tree***. A rooted tree in which the root has degree 1 or 2 and every other vertex has degree ≤ 3 is called a ***binary tree***. These trees are extensively used for data representation. In general, an m-***ary tree*** is a rooted tree in which the root has degree $\leq m$ and every other vertex has degree $\leq m+1$. The vertices of degree 1, other than the root are called the ***leaves***. The largest distance between the root and a leaf is called the ***height of*** T.

A complete m-ary tree is a m-ary tree in which (i) the root has degree m, (ii) every other vertex has degree $m+1$ or 1 and (iii) all the leaves have the same distance from the root.

---

15. Let T be a rooted binary tree on n vertices in which exactly one vertex has degree 2 and every other vertex has degree 1 or 3. Show that: (a) n is odd, and (ii) the number of vertices of degree one is $\dfrac{n+1}{2}$.

16. Find the number of vertices in a complete m-ary tree.

17. Let T be an m-ary tree with root v of degree m and every other vertex of degree $m+1$ or 1. Show that the height of T with k leaves is at least $\log_m k$.

18. Show that the complete binary tree on 7 vertices is not a subgraph of the graph of the 3-cube Q_3.

19. A rooted binary tree T is said to be ***height-balanced***, if for every vertex $v \in V(T)$, the heights of the subtrees rooted at the left and right child of v differ by at most one; the difference is denoted by $b_T(v)$. So $b_T(v) = 0$ or 1. By convention, $b_T(v) = 0$, if v is a leaf.
    (a) Draw all the non-isomorphic height balanced trees of height $h = 0, 1, 2, 3$.
    (b) Find the number of vertices in a height balanced tree T with height h such that $b_T(v) = 0$, for every $v \in V(T)$.
    (c) An height-balanced tree T with $b_T(v) = 1$, for every non-leaf vertex $v \in V(T)$, is called the Fibonacci tree. If n_h denotes the number of vertices in a height balanced tree of height h. Show that $n_0 = 1$, $n_1 = 2$ and $n_h = 1 + n_{h-1} + n_{h-2}$, for $h \geq 2$.

20. Let G be a graph shown in Figure 3.14.

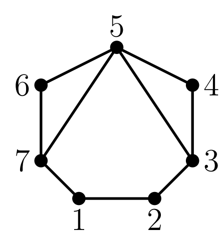


Figure 3.14: G

Draw the following spanning trees of G.
    (i) A spanning tree which is a complete binary tree.
    (ii) A spanning tree of diameter 5.
    (iii) A spanning tree T such that $dist_T(1, v) = dist_G(1, v)$, for every vertex v.

21. Which of the following spanning trees of $K_3 + K_4^c$ exist? Justify your answer.
    (i) A spanning tree which is a complete binary tree.
    (ii) A spanning tree of diameter 3.
    (iii) A spanning tree of diameter 5.

22. Let G be the graph shown in Figure 3.15. Draw a spanning tree T of G such that $dist_T(x, y) \leq 3 \cdot dist_G(x, y)$, for any two vertices x and y.

23. Prove or disprove: The two trees T_1 and T_2 with Prüfer codes $(1, 2, 1, 2, 1, 2)$ and $(1, 1, 1, 2, 2, 2)$ respectively are isomorphic.

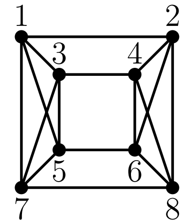


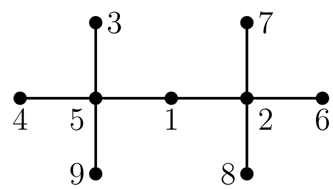
Figure 3.15: G

24. Draw the tree whose Prüfer code is $(6, 4, 1, 1, 5, 3)$.

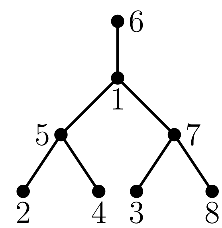
25. What is the tree whose Prüfer code is $(k, k, \dots k)$, for some integer k.

26. What is the tree whose Prüfer code is $(a, b, a, b, \dots, a, b)$, for some integers a and b.

27. Find the Prüfer codes of the following trees.



(a) T_1



(b) T_2

28. (a) If a tree T has diameter 2, find the number of vertices of degree 1.
    (b) If a tree T has diameter 3, find the number of vertices of degree 1.
    (c) Let $n_1(T)$ denote the number of vertices of degree 1 in a tree T, and let $\mathcal{F}_4$ denote the family of all trees on n vertices with diameter 4. Find $\max\{n_1(T) : T \in \mathcal{F}_4\}$.

29. Find all k for which there exists a spanning tree of diameter k of the graph shown below (Figure 3.16):

30. Let $S \subseteq V(G)$ be such that $G - S$ is acyclic. Show that $|S| \geq \dfrac{m - n + 1}{\Delta - 1}$.

31. For which values of k, does there exist an acyclic graph with 17 vertices, 8 edges and k components? Draw two non-isomorphic graphs with these properties.

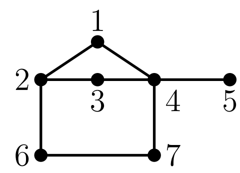


Figure 3.16: G

32. Let T be a tree on 21 vertices with degree sequence $(1^{15}, 3^k, 5^t, 6^1)$. Find k and t.

33. Show that there exists a tree with degree sequence $(1^t, 2, 3, \ldots, n)$ on $t + n - 1$ vertices iff $t - 3 = \dfrac{n(n-3)}{2}$.

34. Draw all the non-isomorphic trees which are isomorphic with their complements. Justify that you have drawn all the non-isomorphic trees.

35. Find the number of non-isomorphic trees with $n(\geq 6)$ vertices and maximum degree $n - 3$.

36. For which values of k, an acyclic graph with 17 vertices, 8 edges and k components exists? Draw two non-isomorphic graphs with these properties.

37. Show that every connected graph G contains a spanning tree T such that $diam(T) \leq 2 \cdot diam(G)$.

38. Show that a connected graph G is a tree iff $G+e$ contains a unique cycle, where e is any new edge.

39. Let G be a connected graph. Show that G contains a connected subgraph on k vertices, for every k, $1 \leq k \leq n$.

40. Let G be a connected graph and let $F \subseteq E(G)$. Show that there exists a spanning tree of G containing F iff $[F]$ is acyclic.

41. Let G be a connected graph and $S \subseteq V(G)$. Show that there exists a spanning tree T of G such that $deg_T(x) = deg_G(x)$, for every $x \in S$ iff $[S]$ and $[S, V - S]$ are acyclic.

42. Show that G is a forest iff every induced subgraph of G contains a vertex of degree ≤ 1.

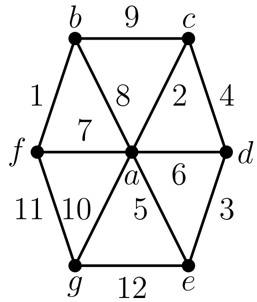
43. By using Kruskal' algorithm compute a minimum weight spanning tree of the graph whose weighted adjacency matrix is given below. Show the tree generated at every step of the algorithm.

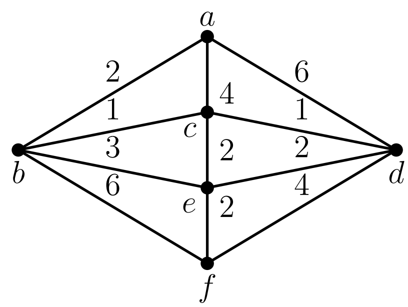
	a	b	c	d	e	f
a	0	10	6	0	0	0
b	10	0	5	0	3	2
c	6	5	0	4	0	1
d	0	0	4	0	4	0
e	0	3	0	4	0	7
f	0	2	1	0	7	0

44. Show that the following "dual" of Kruskal's algorithm outputs a minimum spanning tree.
Input: A weighted connected graph (G, $\mathcal{W}$).
Output: A minimum spanning tree T_k of G.
Step 1(Initial): Arrange the edges of G in non-increasing order, say π. Define H_0 to be the graph G. (During the execution of the algorithm the edges are deleted to yield T_k.)
Step 2: Select the first edge from π say e_1 such that $H_0 - e_1$ is connected and call the resulting graph H_1; if there is no such edge go to step 4.
Step 3 (Recursion): After deleting i edges say $e_1, e_2, \ldots, e_i$, and forming the graph H_i, find the next edge say e_{i+1} in the subsequence succeeding e_i in π such that $H_i - e_{i+1}$ is connected and call the resulting graph H_{i+1}; if there is no such edge go to step 4.
Step 4 (Termination): Stop when $m-n+1$ edges are deleted from G. Output H_{m-n+1} as T_k.

45. Find a minimum spanning tree of the weighted graph shown below by using Prim's algorithm with g as the starting vertex. Draw all the intermediate trees which lead to MST. What is the total weight of MST?

46. Repeat the above problem for the graph shown below.





47. (a) Give an example of a tree T on 6 vertices such that T^c is disconnected.

    (b) Characterize the trees whose complements are disconnected.

48. Let $r, s(\geq 2)$ be integers. Given an example of a simple graph G on $(r-1)(s-1)$ vertices such that G contains no tree on r vertices and G^c contains no K_s.

49. A ***star*** is a tree on n vertices with one vertex of degree $n-1$ and $n-1$ vertices of degree 1. If T is a tree on $n(\geq 1)$ vertices which is not a star, then show that T^c contains an isomorphic copy of T.

Module 4 Special classes of graphs

Contents

There are many classes of graphs which are interesting at an elementary level and also at an advanced level. In this chapter, we introduce three such classes. Before that we define two general concepts.

Definitions.

(1) A graph theoretic property P is said to be an ***hereditary property****, if G has property P, then every subgraph of G has property P.*

(2) A graph theoretic property P is said to be an ***induced hereditary property****, if G has property P, then every* ***induced subgraph*** *of G has property P.*

Remarks.

- If G is acyclic, then every subgraph of G is acyclic. Therefore, acyclicity is an hereditary property. It is also induced hereditary.
- If G is connected, then every subgraph of G need not be connected. Therefore, connectedness is not a hereditary property.
- If P is hereditary, then it is induced hereditary. However, the converse is not true. For example, if G is complete, then every subgraph of G need not be complete but every induced subgraph of G is complete.

In the context of induced hereditary properties, the following terminology is useful.

Definitions. *Let H be a graph and $\mathcal{F}$ be a family of graphs.*

(1) A graph G is said to be H-free, if G contains no induced subgraph isomorphic to H.

(2) A graph G is said to be $\mathcal{F}$-free if G contains no induced subgraph isomorphic to any graph in $\mathcal{F}$.

For example, the graph shown in Figure 4.1 is C_4-free but it is not P_4-free. However, it contains a C_4.

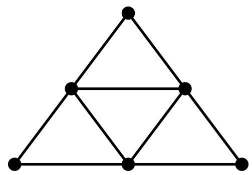


Figure 4.1: A C_4-free graph.

4.1 Bipartite Graphs

Immediate generalization of trees are bipartite graphs.

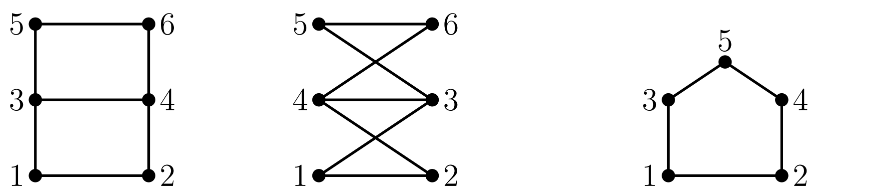
Definition. *A graph G is said to be **bipartite**, if $V(G)$ can be partitioned into two parts A and B such that*

(i) no two vertices in A are adjacent, and

(ii) no two vertices in B are adjacent.

To emphasize, the two parts A and B, we denote a bipartite graph G by $G[A, B]$.

A bipartite graph and a non-bipartite graph are shown in Figure 4.2.



(a) A bipartite graph and its bipartition (b) A non-bipartite graph

Figure 4.2: Bipartite and non-bipartite graphs.

Clearly,

(i) If G is bipartite and $H \subseteq G$, then H is bipartite. Equivalently, if H is non-bipartite and $H \subseteq G$, then G is non-bipartite. Therefore, "bipartiteness" is a hereditary property.

(ii) Every odd cycle is non-bipartite.

Therefore, if G is bipartite, then it contains no odd cycles. The following characterization of bipartite graphs says that the converse too holds.

Theorem 4.1. *A graph G is a bipartite graph if and only if it contains no odd cycles.*

Proof: (1) G is bipartite $\Rightarrow G$ contains no odd cycles.
Let G be a bipartite graph with bipartition $[A, B]$. Let $C = (v_1, v_2, \ldots, v_k, v_1)$ be a k-cycle in G. The vertices v_i belong alternately to A and B. Let, without loss of generality, $v_1 \in A$. Then, it follows that $v_j \in A$ iff j is odd and $v_j \in B$ iff j is even. Since $v_1 \in A$ and $(v_1, v_k) \in E$, we deduce that $v_k \in B$. So k is even.
(2) G has no odd cycles $\Rightarrow G$ is bipartite.
We give two proofs.

First proof: Clearly, a graph is bipartite iff each of its components is bipartite. So, it is enough if we prove the theorem for connected graphs. Let x be a vertex in G. Define the following sets:

$$A = \{v \in V : dist(x, v) \text{ is even}\} \text{ and } B = \{v \in V : dist(x, v) \text{ is odd}\}.$$

Clearly, $A \cap B = \phi$ and $A \cup B = V(G)$. We claim that no two vertices in A are adjacent. On the contrary suppose that $v_1, v_2 \in A$ are adjacent. So, $dist(x, v_1)$ is even, say $2r$, and $dist(x, v_2)$ is even, say $2s$. Let $P_1(x, v_1)$ and $P_2(x, v_2)$ be the paths of length $2r$ and $2s$ respectively. Let z be the last vertex common between $P_1(x, v_1)$ and $P_2(x, v_2)$. See Figure 4.3.

Since P_1 and P_2 are of minimum length it follows that the subpaths $P_1(x, z)$ and $P_2(x, z)$ have the same length, say k. Then the sequence $P_1(z, v_1), (v_1, v_2), P_2(v_2, z)$ is a cycle of length $(2r-k)+1+(2s-k)$, which is an odd integer. This is a contradiction,

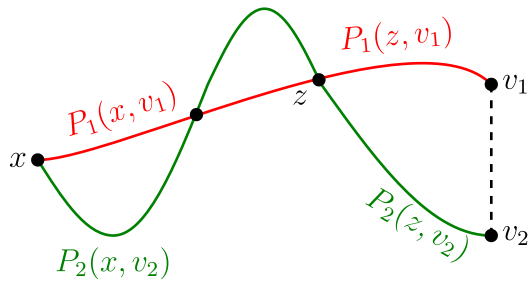


Figure 4.3: $(P(z, v_1), (v_1, v_2), P(v_2, z))$ is a cycle of odd length.

since, G has no odd cycles. Hence, no two vertices in A are adjacent. Similarly, no two vertices in B are adjacent. Hence, $[A, B]$ is a bipartition of G.

Second Proof: By induction on m.

Basic Step: If $m = 0$ or 1, then obviously G is bipartite.

Induction Step: Assume that every graph with $m - 1$ edges and with no odd cycles is bipartite. Let G have m edges and let (u, v) be an edge in G. Consider the graph $H = G - (u, v)$. Since G has no odd cycles, H too has no odd cycles. Therefore, by induction hypothesis, H is bipartite. We now make two cases and in each case obtain a bipartition of G.

Case 1: u and v are connected in H.

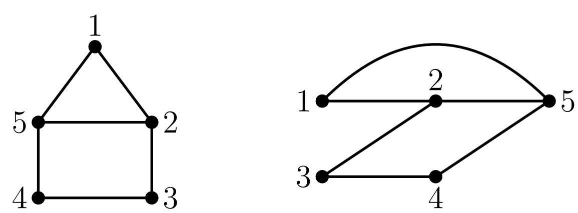
Let $[A, B]$ be a bipartition of H. We claim that u and v are in different sets A and B. On the contrary, suppose that u and v belong to the same set, say A. Since, u and v are connected, there is a path $P(u, v) = (u = u_1, u_2, \ldots, u_{k-1}, u_k = v)$. The vertices u_i belong alternately to A and B and moreover the terminal vertices u and v belong to A; hence the number of vertices in P is odd. Therefore, $P(u, v)$ is a path of even length. But then $(P(u, v), v, u)$ is cycle of odd length in G; a contradiction. So our claim holds; let $u \in A$ and $v \in B$. But then $[A, B]$ is bipartition of $H + (u, v) = G$ too.

Case 2: u and v are not connected in H.

Let D be the component in H which contains u. Since H is bipartite, both D and $H - V(D)$ are bipartite. Let $[A_1, B_1]$ be a bipartition of D and $[A_2, B_2]$ be a bipartition of $H - V(D)$. We can assume that $u \in A_1$ and $v \in B_2$ (otherwise, suitably relabel A_i and B_i). Then $[A_1 \cup A_2, B_1 \cup B_2]$ is a bipartition of G. $\square$

The concept of bipartite graphs can be straightaway generalized to k-partite graphs.

Definition. *A graph G is said to be k-partite, if there exists a partition $(V_1, V_2, \ldots, V_k)$ of $V(G)$ such that no two vertices in V_i $(1 \leq i \leq k)$ are adjacent. It is denoted by $G[V_1, V_2, \ldots, V_k]$.*



(a) G (b) A 3-partition of G.

Figure 4.4: A 3-partite graph G. Note that it is not a bipartite graph.

While bipartite graphs were easily characterized, the characterization of k-partite graphs $(k \geq 3)$ has remained open.

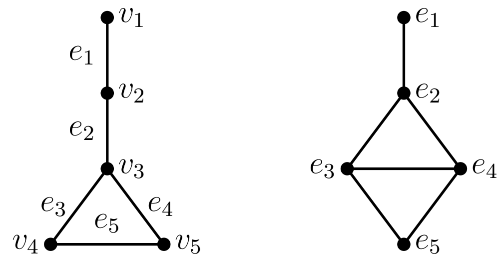
Definition. *A k-partite graph $G[V_1, V_2, \ldots, V_k]$ is said to be a complete k-partite graph if $(x, y) \in G$, whenever $x \in V_i, y \in V_j, i \neq j$. It is denoted by $K_{n_1, n_2, \ldots, n_k}$, if $|V_i| = n_i$, $1 \leq i \leq k$.*

4.2 Line Graphs (Optional)

Given a graph G, we can "derive" many graphs based on G. One such family of graphs called "line graphs" is extensively studied.

Definition. *Let H be a simple graph with n vertices $v_1, v_2, \ldots, v_n$ and m edges $e_1, e_2, \ldots, e_m$. The* ***line graph*** *$L(H)$ of H is a simple graph with the vertices $e_1, e_2, \ldots, e_m$ in which e_i and e_j are adjacent iff they are adjacent in H.*

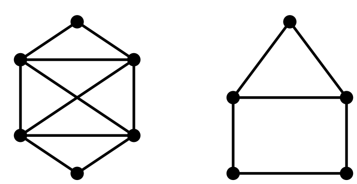
A graph and its line graph are shown below.



(a) H (b) $L(H)$

Figure 4.5: A graph and its line graph.

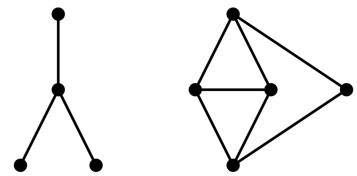
- A graph G is said to be a ***line graph***, if there exists a simple graph H such that $L(H) = G$. The graph shown in Figure 4.5b is a line graph. Two more graphs G_1 and G_2 which are line graphs are shown below. Find simple graphs H_1 and H_2 such that $G_1 = L(H_1)$ and $G_2 = L(H_2)$.



(a) G_1 (b) G_2

Figure 4.6: Two line graphs.

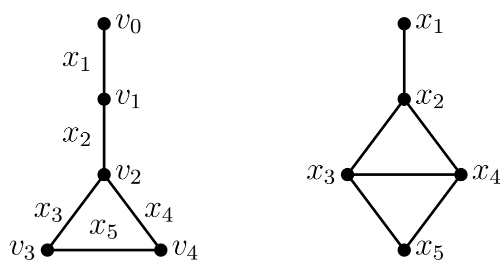
- Two graphs which are not line graphs are shown in Figure 4.7. That is, one can show that there are no graphs H_3 and H_4 such that $G_3 = L(H_3)$ and $G_4 = L(H_4)$, either with case-by-case analysis or by looking at all the graphs on four edges and five edges, respectively. We leave this exercise to the reader.



(a) G_3 (b) G_4

Figure 4.7: Two graphs which are not line graphs.

In view of the existence of graphs which are line graphs and also the existence of graphs which are not line graphs, a natural question arises: Find necessary and sufficient conditions for a graph to be a line graph. Towards this characterization, let us look at the graphs shown in Figure 4.8 more closely.



(a) H (b) G

Figure 4.8: $G = L(H)$

We observe that

$$Q_0 = [\{x_1\}], Q_1 = [\{x_1, x_2\}], Q_2 = [\{x_2, x_3, x_4\}], Q_3 = [\{x_3, x_5\}], Q_4 = [\{x_4, x_5\}]$$

are complete subgraphs of G having the following properties:

(i) Any edge of G belongs to exactly one complete subgraph.

(ii) Any vertex of G belongs to at most two complete subgraphs.

Notice that the vertices of Q_i $(i = 0, 1, 2, 3, 4)$ are the edges incident with the vertex v_i $(i = 0, 1, 2, 3, 4)$ in H. These are the key observations to obtain the first characterization of line graphs. Clearly a graph G is a line graph iff each component of G is a line graph. Moreover, it can be easily verified that every graph on at most 2 vertices is a line graph. So, in the following we assume that G is connected and $n \geq 3$.

Theorem 4.2 (Krausz, 1943)**.** *A connected graph G is a line graph if and only if there exists a family F of complete graphs $\{Q_1, Q_2, \ldots, Q_p\}$ in G such that*

(i) any edge of G belongs to exactly one complete subgraph, and

(ii) any vertex of G belongs to at most two complete subgraphs.

Proof. (1) G is a line graph $\Rightarrow$ G contains a family F of complete subgraphs as described in the theorem.

Since G is a line graph, there exists a simple graph H such that $G = L(H)$.

If v is a vertex of H, then the set of edges $\{x_1, x_2, \ldots, x_d\}$ incident with v in H induces a complete subgraph $Q_v = [\{x_1, x_2, \ldots, x_d\}]$ in G. Our claim is that the family of complete subgraphs $F = \{Q_v \subseteq G : v \in H\}$, satisfies (i) and (ii).

Claim 1: F satisfies (i).

Let $(x, y) \in E(G)$. Then $x = (u, v)$ and $y = (v, w)$ for some vertices u, v, w in H. So, $(x, y) \in E(Q_v)$ in G and no other complete subgraph.

Claim 2: F satisfies (ii).

Let x be a vertex of G; so x is an edge in H. Let $x = (u, v)$, where $u, v \in V(H)$. Then $x \in V(Q_u) \cap V(Q_v)$ and no other complete subgraph in F.

(2) G contains a family $F = \{Q_1, Q_2, \ldots, Q_p\}$ of complete subgraphs as described in the theorem $\Rightarrow$ G is a line graph.

We construct a graph H such that $L(H) = G$. For each Q_i we take a vertex v_i, $i = 1, 2, \ldots, p$. Next, for each vertex x_i of G which belongs to exactly one of the complete subgraphs of F, we take a vertex u_i, $i = 1, 2, \ldots, r$ (say). Then $u_1, u_2, \ldots, u_r$

and $v_1, v_2, \ldots, v_p$ are the vertices of H. Two of these vertices are joined in H iff the corresponding complete graphs share a vertex in G. Then $L(H) = G$. $\square$

We next state two structural characterizations of line graphs. The graph shown in Figure 4.9 is called a ***diamond***.

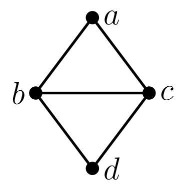


Figure 4.9: A diamond, $K_4 - e$.

A diamond D in G with triangles $[a, b, c]$ and $[b, c, d]$ as shown above is called an odd diamond of G, if

(i) $D \sqsubseteq G$,

(ii) there is a vertex $x \in V(G) - V(D)$ which is adjacent with odd number of vertices of $[a, b, c]$, and

(iii) there is a vertex $y \in V(G) - V(D)$ which is adjacent with odd number of vertices of $[b, c, d]$;

Here the vertices x and y need not be distinct. For example, the three graphs shown in Figure 4.10 contain odd diamonds.

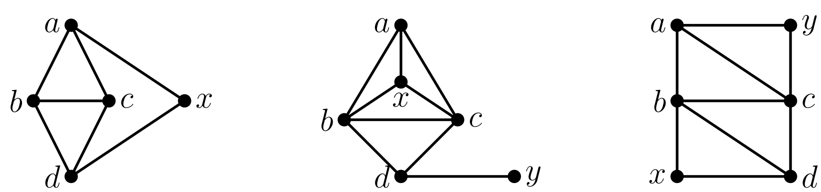
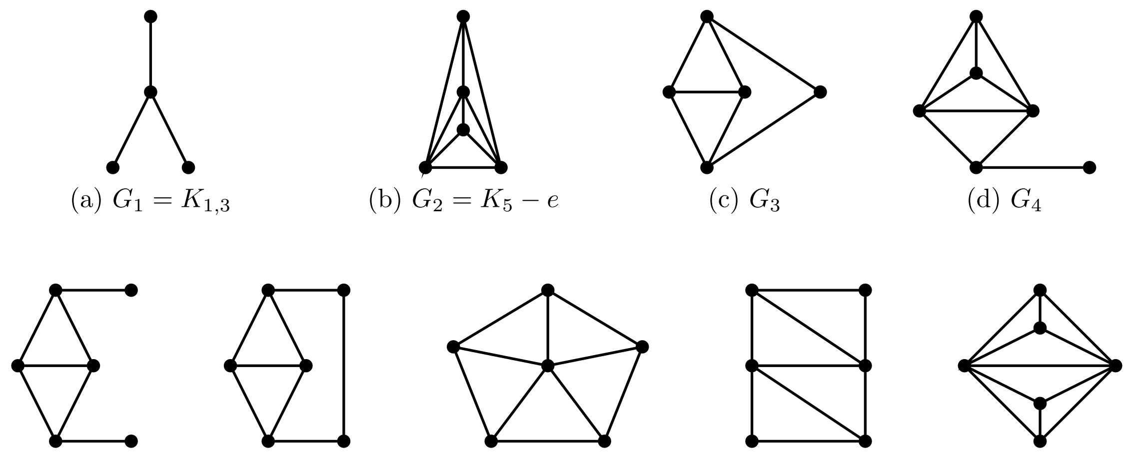


Figure 4.10: Graphs containing odd diamonds (Can you identify them?)

Theorem 4.3 (A.V. Van Rooiz and H.S. Wilf, 1965)**.** *A graph G is a line graph iff it contains no induced $K_{1,3}$ and no odd diamonds.*

Theorem 4.4. *(Beineke, 1968) A graph G is a line graph iff it contains none of the nine graphs shown in Figure 4.11 as an induced subgraph.*



(a) $G_1 = K_{1,3}$ (b) $G_2 = K_5 - e$ (c) G_3 (d) G_4

(e) G_5 (f) G_6 (g) G_7 (h) G_8 (i) G_9

Figure 4.11: Nine forbidden graphs of line graphs

4.3 Chordal Graphs (Optional)

Definitions.

- *Let $C_k (k \geq 4)$ be a cycle in G. An edge of G which joins two non-consecutive vertices of C_k is called a **chord**.*
- *A graph G is said to be a **chordal graph**, if every cycle $C_k (k \geq 4)$ in G has a chord. A chordal graph is also called a **triangulated graph**.*

Remarks.

1. Every induced subgraph of a chordal graph is chordal.
2. A subgraph of a chordal graph need not be a chordal graph.

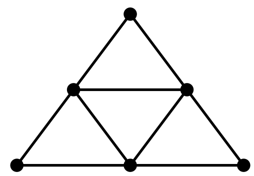


Figure 4.12: A chordal graph and a non-chordal graph

The class of chordal graphs and its subclasses have been a subject of considerable interest because of their structural properties and interesting characterizations. We state and prove two characterizations.

Definitions. *Let a, b be any two non-adjacent vertices in a graph G.*

(i) An ***a-b-separator*** *is a subset S of vertices such that a and b belong to two distinct components $G - S$.*

(ii) A ***minimal a-b-separator*** *S is an a-b-separator such that no proper subset of S is an a-b-separator.*

In the graph shown in Figure 4.13, $\{u, v, x\}$ is a y-z-separator. It is not a minimal y-z-separator, since $\{u, x\}$ is a y-z-separator. In fact, $\{u, x\}$ is a minimal y-z-separator. However, $\{u, v, x\}$ is a minimal y-w-separator. These examples illustrate that the minimality of an a-b-separator depends on the vertices a and b. Next, $\{z\}$ is a minimal w-p-separator and $\{u, x\}$ is also a minimal w-p-separator. These examples illustrate that the minimal separators need not have same number of vertices.

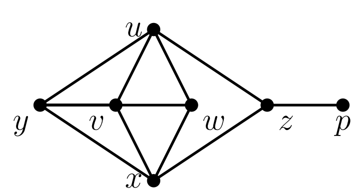


Figure 4.13: Illustration for separators

Remark.

- For any two vertices x, y in G, there exists a x-y-separator if and only if x and y are non-adjacent.

Theorem 4.5 (Dirac, 1961)**.** *A graph is chordal iff every minimal a-b-separator induces a complete subgraph, for every pair of non-adjacent vertices a and b in G.*

Proof. (1) Every minimal a-b-separator, for every pair of non-adjacent a, b in G, induces a complete subgraph in $G \Rightarrow G$ is chordal.

Let $C : (x_1, x_2, \ldots, x_r, x_1)$ be a cycle in G of length ≥ 4; see Figure 4.14.

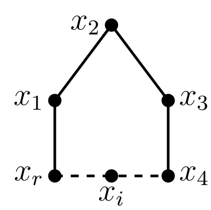


Figure 4.14: $(x_1, x_3) \in E(G)$ or $(x_2, x_i) \in E(G)$.

If (x_1, x_3) is an edge, then it is a chord. Next suppose $(x_1, x_3) \notin E(G)$ and let S be a minimal x_1-x_3-separator. Then S contains x_2 and some x_i $(4 \leq i \leq r)$. Since S induces a complete subgraph we deduce that $(x_2, x_i) \in E(G)$ and it is a chord.

(2) G is chordal $\Rightarrow$ Every minimal a-b-separator, for every pair of non-adjacent a, b in G, induces a complete subgraph in G.

Let S be a minimal a-b-separator in G. If $|S| = 1$, then $[S] = K_1$, and so the assertion holds. If $|S| \geq 2$, let $x, y \in S$. Our goal is to show that $(x, y) \in E(G)$; see Figure 4.15. Let G_a and G_b be the components of $G - S$ containing a and b respectively.

Since S is minimal, the sets of edges $[x, V(G_a)], [y, V(G_a)], [x, V(G_b)]$ and $[y, V(G_b)]$ are all non-empty. Let $(x, a_1, a_2, \ldots, a_r, y)$ be a path in G such that

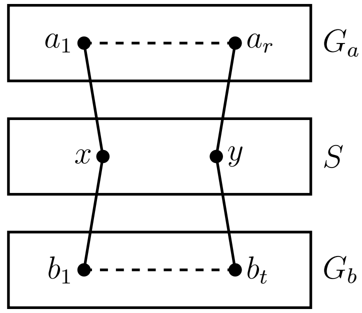


Figure 4.15: Schematic representation of G.

$a_1, a_2, \ldots, a_r \in V(G_a)$ and r is minimum. Let $(x, b_1, b_2, \ldots, b_t, y)$ be a path in G such that $b_1, b_2, \ldots, b_t \in V(G_b)$ and t is minimum.

Consider the cycle $C = (x, a_1, a_2, \ldots, a_r, y, b_t, b_{t-1}, \ldots, b_1, x)$. Since r is minimum, we deduce that

(i) x is non-adjacent to $a_2, a_3, \ldots, a_r$.

(ii) y is non-adjacent to $a_1, a_2, a_3, \ldots, a_{r-1}$.

(iii) a_i, a_j are non-adjacent for every $i, j \in \{1, 2, \ldots, r\}, j \notin \{i+1, i-1\}$; that is, the path $(a_1, \ldots, a_r)$ has no chords.

Similarly, since t is minimum, we deduce that

(i) x is non-adjacent to $b_2, b_3, \ldots, b_t$.

(ii) y is non-adjacent to $b_1, b_2, \ldots, b_{t-1}$.

(iii) b_i, b_j are non-adjacent for every $i, j \in \{1, 2, \ldots, t\}, j \notin \{i+1, i-1\}$; that is, the path $(b_1, \ldots, b_t)$ has no chords.

Since G_a and G_b are components of $G - S$, $(a_i, b_j) \notin E(G)$ for all i, j, $1 \leq i \leq r, 1 \leq j \leq t$. All these non-adjacency properties imply that $(x, y) \in E(G)$, since G is chordal and C is a cycle. Since x and y are arbitrary, it follows that S induces a complete subgraph in G. $\square$

The second characterization of chordal graphs has proved useful in designing algorithms to find various parameters of chordal graphs. Before stating this characterization, we define a new concept and prove a crucial theorem to obtain the characterization.

Definition. *A vertex v of a graph G is called a* ***simplicial vertex****, if its neighborhood $N(v)$ induces a complete subgraph in G.*

In the following graph, x is a simplicial vertex while y and z are non-simplicial vertices.

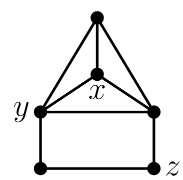


Figure 4.16: A graph with simplicial vertices and non-simplicial vertices.

Theorem 4.6. *Every chordal graph G has a simplicial vertex. Moreover, if G is not complete, then G has two non-adjacent simplicial vertices.*

Proof. We prove the theorem by induction on n. If $n = 1, 2$ or 3, the statement is obvious. So we proceed to the induction step. If G is complete, then every vertex of G is simplicial. Next suppose that G has two non-adjacent vertices say a and b. Let S be a minimal a-b-separator. $[S]$ is complete by the previous theorem. Let G_a, G_b be the components in $G - S$ containing a and b respectively. Let $A = V(G_a)$ and $B = V(G_b)$.

We apply induction hypothesis to the induced subgraph $H = [A \cup S]$ which is chordal. If H is complete, then every vertex of H is a simplicial vertex of H. If H is not complete, then by induction hypothesis, H contains two non-adjacent vertices which are simplicial. One of these vertices is in A, since S is complete. So, in either case A contains a simplicial vertex of H, say x. Since $x \in A$, $N_G(x) = N_H(x)$. So, x

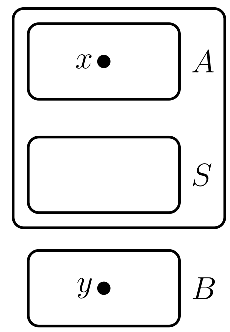


Figure 4.17: Schematic representation.

is also a simplicial vertex of G. Similarly, B contains a simplicial vertex of G, say y. Then x and y are two non-adjacent simplicial vertices of G. $\square$

Definition. *A graph G on n vertices is said to have a* ***perfect elimination ordering (PEO)*** *if its vertices can be ordered $(v_1, v_2, \ldots, v_n)$ such that v_i is a simplicial vertex of the induced subgraph $[\{v_i, v_{i+1}, \ldots, v_n\}]$, $i = 1, 2, \ldots, n-1$.*

In the following graph $(v_1, v_3, v_5, v_2, v_4, v_6)$ is a PEO but $(v_1, v_2, v_3, v_5, v_4, v_6)$ is not a PEO.

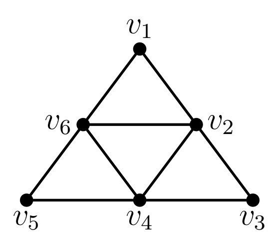


Figure 4.18: A graph with perfect elimination ordering.

Theorem 4.7 (Dirac 1961, Fulkerson and Gross, 1965, Rose 1970)**.** *A graph G is chordal iff it admits a PEO.*

Proof. (1) G is chordal $\Rightarrow$ G admits a PEO.

Since G is chordal, it contains a simplicial vertex, say v_1. Since $G_1 = G - v_1$ is chordal, it contains a simplicial vertex, say v_2. Similarly, $G_2 = G - v_1 - v_2$ contains a simplicial vertex, say v_3. Continuing this process, we obtain an ordering of the vertices $(v_1, v_2, \ldots, v_n)$ which is a PEO of G.

(2) G admits a PEO $\Rightarrow$ G is chordal.

Let $(v_1, v_2, \ldots, v_n)$ be a PEO of G and let C be an arbitrary cycle in G of length ≥ 4. Let v_k be the first vertex in the PEO which belongs to C. Let x and y be the vertices of C adjacent to v_k; see Figure 4.19.

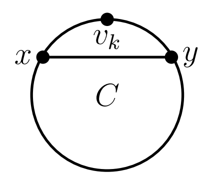


Figure 4.19: C contains the chord (x, y).

By our choice of v_k, it follows that $x, y \in \{v_{k+1}, v_{k+2}, \ldots, v_n\}$. Since v_k is a simplicial vertex, all its neighbors in $\{v_k, v_{k+1}, \ldots, v_n\}$ are mutually adjacent. So, we conclude that x and y are adjacent. Hence, every cycle in G contains a chord. □

Exercises

1. Which of the graphs shown in Figure 4.20 are bipartite? If bipartite, then redraw them with a suitable bipartition.

2. Show that the maximum number of edges in a bipartite graph is $[\frac{n^2}{4}]$. Characterize the bipartite graphs $G[A, B]$ with $[\frac{n^2}{4}]$ edges.

3. If G is a connected bipartite graph, show that the bipartition of $V(G)$ is unique, in the sense that if $[A, B]$ and $[C, D]$ are bipartitions of G, then either (i) $A = C$ and $B = D$ or (ii) $A = D$ and $B = C$.

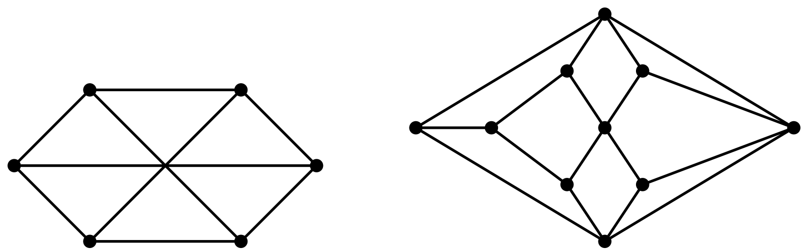


Figure 4.20

4. If G is a connected simple graph such that $G-v$ is bipartite for every $v \in V(G)$, then show that G is either bipartite or G is an odd cycle.

5. If $G[A, B]$ is a k-regular bipartite graph ($k \geq 1$), then show that $|A| = |B|$.

6. If G and H are bipartite, show that $G \square H$ is bipartite.

7. Show that the hypercube Q_d is bipartite, for every $d \geq 1$.

8. Draw all the simple bipartite graphs G such that G^c is also bipartite.

9. A bipartite simple graph $G[A, B]$ has 22 vertices where $|A| = 12$. Suppose that every vertex in A has degree 3, and every vertex in B has degree 2 or 4. Find the number of vertices of degree 2.

10. Show that a simple graph G is a complete k-partite graph for some $k(\geq 2)$ iff whenever $(u, v) \notin E(G)$ and $(v, w) \notin E(G)$, then $(u, w) \notin E(G)$. (Hint: Consider G^c.)

11. Find the degree sequence and the number of edges of $K_{1,2,\ldots,r}$ and $(K_{1,2,\ldots,r})^c$.

12. Find
    (a) $k_0(K_{1,2,\ldots,r})$ and $k_1(K_{1,2,\ldots,r})$.
    (b) $k_0(K_{p,p,\ldots,p})$ and $k_1(K_{p,p,\ldots,p})$, where p is repeated k times.
    (c) $k_0(K_{a,b,c})$ and $k_1(K_{a,b,c})$, where $1 \leq a \leq b \leq c$.

13. Find the diameter and the girth of $K_{1,2,\ldots,p}$.

14. Suppose that the average degree of a simple graph G is b and x is a vertex of G. Show that $G - x$ has average degree at least b iff $deg_G(x) \leq \frac{b}{2}$. (Average degree of a graph $H := \frac{2m(H)}{n(H)}$.)

15. Let $k \geq 2$. Show that there is no bipartite graph on n vertices with $n-1$ vertices of degree k and one vertex of degree $k-1$.

16. If G is a simple graph on 7 vertices, then show that either $C_3 \subseteq G$ or $C_4 \subseteq G^c$. Construct a simple graph H on 6 vertices such that neither $C_3 \subseteq H$ nor $C_4 \subseteq H^c$.

17. Let $\underline{a} = (a_1 \geq a_2 \geq \cdots \geq a_r)$ and $\underline{b} = (b_1 \geq b_2 \geq \cdots \geq b_s)$ be sequences of non-negative integers such that $\sum_{i=1}^{r} a_i = \sum_{j=1}^{s} b_j$. Show that there exists a simple bipartite graph $G[\{x_1, x_2, \ldots, x_r\}, \{y_1, y_2, \ldots, y_s\}]$ such that $deg(x_i) = a_i$, $1 \leq i \leq r$ and $deg(y_j) = b_j$, $1 \leq j \leq s$ iff

$$\sum_{i=1}^{r} \min\{a_i, k\} \geq \sum_{j=1}^{k} b_j, \text{ for } k = 1, 2, \ldots, s.$$

18. Draw a 3-partite graph on 7 vertices with maximum number of edges.

19. Prove or disprove: If $K_{r,s,t}$ is a regular graph, then $r = s = t$.

20. Draw a spanning tree T of $G = K_{4,4}$ such that $dist_T(u, v) \leq 3 \cdot dist_G(u, v)$, for every pair of vertices u, v.

21. Draw the following spanning trees of $K_{2,3}(= G)$.
    (a) A spanning tree which is a complete binary tree.
    (b) A spanning tree of diameter 3.
    (c) A spanning tree T of G such that $dist_T(x, y) \leq dist_G(x, y) + 1$, for every pair of vertices x, y.

22. If G is a tree with maximum degree 3 and bipartition $[X, Y]$, where $|X| \leq |Y|$, then show that there exist at least $|Y| - |X| + 1$ vertices of degree 1 in G.

23. The following $(0, 1)$- matrix represents a bipartite graph G with vertices a, b, c, d in one part and e, f, g, h in another part. Draw G.

	e	f	g	h
a	1	0	1	0
b	0	1	1	0
c	1	0	1	1
d	1	1	1	0

24. Prove or disprove: There is a bipartite graph with bipartition $[X, Y]$, where $X = \{x_1, x_2, x_3, x_4, x_5\}$, $Y = \{y_1, y_2, y_3, y_4, y_5\}$, $deg(x_j) = deg(y_j) = j$ for $j = 1, 2, 4, 5$ and $deg(x_3) = deg(y_3) = 4$.

25. Prove or disprove: If G is a r-regular bipartite simple graph, the G contains a spanning k-regular subgraph H, for every k, $1 \leq k \leq r$.

26. Characterize the trees which are line graphs.

27. Characterize the connected bipartite graphs which are line graphs.

28. Prove or disprove:
    (a) If G is a line graph, then $G - v$ is a line graph, for every $v \in V(G)$.
    (b) If G is such that $G - v$ is a line graph, for every $v \in V(G)$, then G is a line graph.

29. If $e = (u, v) \in E(G)$, then find $deg_{L(G)}(e)$.

30. Characterize the connected graphs G such that $G = L(G)$.

31. If G is p-edge-connected then show the following:
    (a) $L(G)$ is p-vertex-connected.
    (b) $L(G)$ is $(2p - 2)$-edge-connected.

32. For a simple graph $G(n \geq 4)$, show that the following statements are equivalent.
    (a) G is a line graph of a triangle-free graph.
    (b) G is $\{K_{1,3}, K_4 - e\}$-free.

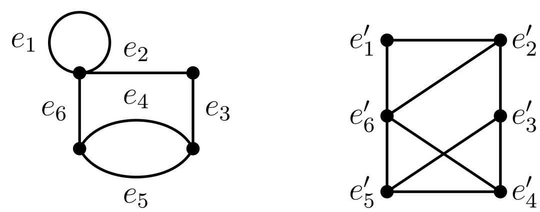
33. For a simple graph $G(n \geq 4)$, show that the following statements are equivalent.
    (a) G is the line graph of a bipartite graph.
    (b) G is $\{K_{1,3}, K_4 - e, C_{2k+1}(k \geq 2)\}$-free.

34. For a simple connected graph G, show that the following statements are equivalent.
    (a) G is the line graph of a tree.
    (b) G is $\{K_{1,3}, K_4 - e, C_k(k \geq 4)\}$-free.

35. Let G be a simple connected graph. Show that $L(G)$ is bipartite iff G is a path or an even cycle.

36. Show that a simple graph G is chordal iff $L(G)$ is chordal.

The line graphs of simple graphs were defined in this chapter. The line graph of any graph (not necessarily simple) can be defined as follows.
Let G be a graph with edges $e_1, e_2, \ldots, e_m$. The line graph $L(G)$ of G has vertices $e'_1, e'_2, \ldots, e'_m$. Two vertices e'_i and e'_j are joined by an edge in $L(G)$ iff e_i and e_j are adjacent in G.
A graph H is said to be a line graph, if there exists a graph G such that $L(G) = H$.



(a) G (b) $L(G)$

Figure 4.21: A graph and its line graph.

---

37. Find which of the graphs shown in Figure 4.11 are line graphs of graphs (which are not necessarily simple).

38. Prove or disprove:
    (a) If G is chordal, then G^c is chordal.
    (b) If G is chordal, then $G - v$ is chordal, for every $v \in V(G)$.
    (c) If G is such that $G - v$ is chordal, for every $v \in V(G)$, then G is chordal.

39. Get a PEO of $K_p^c + K_t$.

Module 5 Eulerian Graphs

Contents

5.1 Motivation and origin

The existence of closed trails containing all the edges in a graph is the subject of first paper in graph theory (1736). It was written by Leonhard Euler (1707−1783)[1], thus initiating the theory of graphs. Like many combinatorial problems, Euler's paper has its motivation in a problem that can be easily stated.

Königsberg-7-bridge-problem: The river Pregel flows through the city of Königsberg (located in Russia) dividing the city into four land regions of which, two are banks and two are islands. During the time of Euler, the four land regions were connected by 7 bridges as shown below.

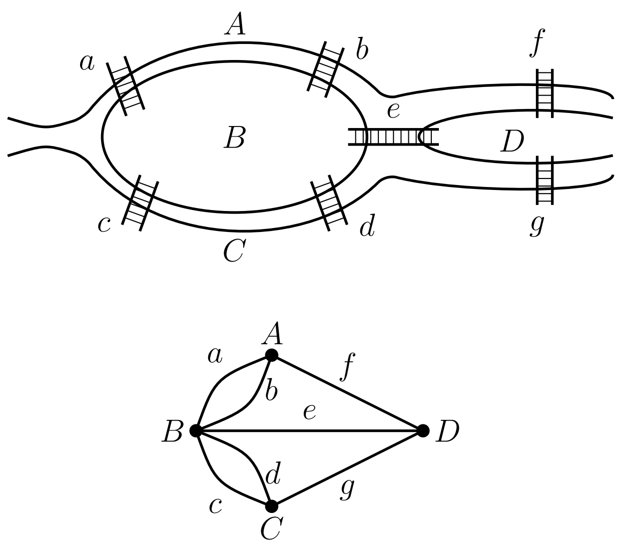


Figure 5.1: Pregel river in the city of Königsberg with 7 bridges and its representation as visualized by Euler.

The citizens of Königsberg had an entertaining exercise. Start from any land region and come back to the starting point after crossing each of the seven bridges

[1]L. Euler: Solutio Problematis ad Geometriam Situs Pertinents [Traslation: The solution of a problem relating to the geometry of position], *Commentar: Academiae scientiarum Imperialis Petropolitanae*, 8(1736), p 128-140.

exactly once. Euler explained that it is impossible to do so by using the terminology of points (representing the land regions) and lines (representing the bridges). Hence, he titled his paper as "Solutions to a problem relating to the geometry of positions." Through this explanation, he laid the foundation for Graph Theory.

Definitions.

- *A trail in a graph which contains all its edges is called an **Eulerian trail**. It can be open or closed.*
- *A graph is called an **Eulerian graph** if it contains a **closed** Eulerian trail.*

Note that an Eulerian graph is necessarily connected. Moreover, if G is Eulerian then one can choose an Eulerian trail starting and ending from any given vertex.

Theorem 5.1 (Euler, 1736)**.** *A connected graph G is Eulerian iff every vertex has even degree.*

Proof. (1) G is Eulerian $\Rightarrow$ Every vertex has even degree.

Let $W = (v_0, e_1, v_1, e_2, v_2, \ldots, v_{t-1}, e_t, v_t(= v_0))$ be a closed Eulerian trail in G. If v_i is an internal vertex ($\neq v_0$) of W appearing k times, then $deg_G(v_i) = 2k$. If v_0 appears r times internally, then $deg_G(v_i) = 2r + 2$.

(2) (Proof due to Fowler, 1988) Every vertex is of even degree $\Rightarrow$ G is Eulerian.

This implication is proved by induction on m. If $m \leq 2$, then G is one of the following four graphs. Clearly, each of them is Eulerian.

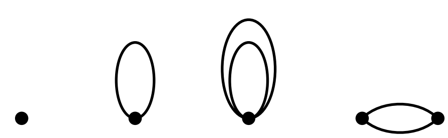


Figure 5.2: All connected graphs with at most 2 edges and every vertex of even degree.

We proceed to prove the induction step assuming that the implication holds for all connected graphs with at most $m-1$ edges and that G has m edges. Let x be any vertex of G, and (x, w_1), (x, w_2) be two of the edges incident with x; w_1, w_2 need not be distinct. Consider the graph H obtained from G by deleting (x, w_1), (x, w_2) and adding a new edge $f = (w_1, w_2)$; see Figure 5.3. The graph H has $m-1$ edges and its every vertex has even degree. However, H may be connected or disconnected. So, we consider two cases.

Case 1: H is connected.

By induction hypothesis, H contains a closed Eulerian trail, say

$$W = (v_0, e_1, v_1, \ldots, w_1, f, w_2, \ldots, v_0).$$

Then the trail

$$W^* = (v_0, e_1, v_1, \ldots, w_1, \underbrace{(w_1, x), x, (x, w_2)}, w_2, \ldots, v_0)$$

is a closed trail in G; see the figure below.

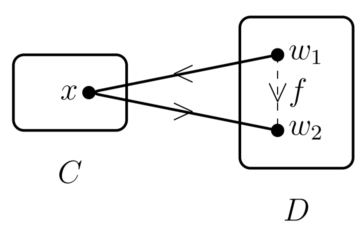


Figure 5.3: Construction of a new graph and the extension of a trail.

Case 2: H is disconnected.

In this case, H contains two components, say C and D such that $x \in C$ and $f \in D$. Both the graphs C and D have less than m edges and every vertex in $V(C) \cup V(D)$ is even. Hence, by induction hypothesis, C and D contain Eulerian trails, say $W_1(x,x)$ and $W_2 = (v_0, e_1, v_1, \ldots, w_1, f, w_2, \ldots, v_0)$. Then

$$(v_0, e_1, v_1, \ldots, w_1, \underbrace{(w_1, x), W_1(x,x), (x, w_2)}, w_2, \ldots, v_0)$$

is an Eulerian trail in G. □

Using the above theorem it is easy to conclude that the graph shown in Figure 5.1 is non-Eulerian.

Corollary. *If a connected graph G contains exactly two vertices of odd degree say x and y, then it contains a (x,y)-Eulerian trail.*

Proof. Let G^* be a new graph obtained by adding a new vertex z and joining it to x and y. Clearly, G^* is an Eulerian graph in which z has degree 2. Without loss of generality, let

$$W = (z, (z,x), x, \ldots, y, (y,z), z)$$

be a closed Eulerian trail in G^*. But then the sub-trail $W' = (x, \ldots, y)$ of W is a required (x,y)-trail in G. □

Corollary. *If a connected graph G contains $2k$ (≥ 2) vertices of odd degree, then $E(G)$ can be partitioned into k sets $E_1, E_2, \ldots, E_k$ such that each E_i induces a trail.*

Proof. Apply the above proof technique. □

5.2 Fleury's algorithm to generate a closed Eulerian trail

Though Euler's theorem neatly characterizes Eulerian graphs, its proof is existential in nature. Fleury(1983) described an algorithm to generate a closed Eulerian trail in a given connected graph in which every vertex has even degree. In the following we describe this algorithm and prove that it indeed generates an Eulerian trail.

Fleury's algorithm:

Input: A weighted connected graph $(G, \mathcal{W})$ in which every vertex has even degree.

Output: A closed Eulerian trail W.

Step 1: Choose a vertex v_0 (arbitrarily) and define the trail $W_0 := v_0$.

Step 2: After selecting a trail, say $W_k = (v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_k, v_k)$, form the graph $G_k = G_{k-1} - e_k$. (That is, $G_k = G - \{e_1, e_2, \ldots, e_k\}$, where $G_0 = G$).

Step 3:

(i) If there is no edge incident with v_k in G_k, then **stop**. Declare W_k is a closed Eulerian trail of G.

(ii) If there is an edge incident with v_k in G_k, select an edge say $e_{k+1} = (v_k, v_{k+1})$, giving preference to a non-cut-edge of G_k. Define $W_{k+1} = (v_0, e_1, v_1, e_2, \ldots, e_k, v_k, e_{k+1}, v_{k+1})$. Goto Step 2 with W_{k+1}.

An illustration: The trails $W_1, \ldots, W_{10}$ and the graphs $G_1, \ldots, G_{10}$ generated by the algorithm are shown below. W_{10} is a closed Eulerian trail. For simplicity, we have shown the trails with edges alone.

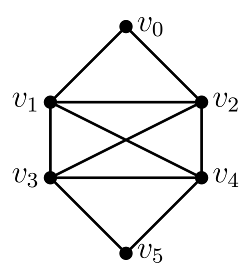
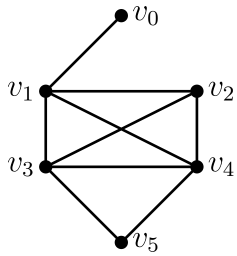
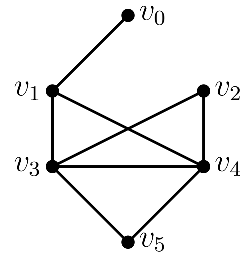


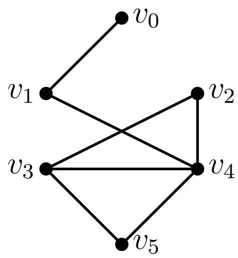
Figure 5.4: Input graph with $W = v_0$ and $G = G_0$.



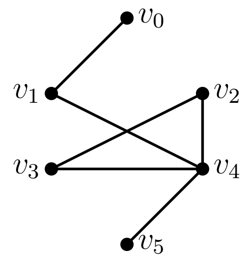
(a) $W_1 = (v_0, v_2)$;
$G_1 = G - \{(v_0, v_2)\}$



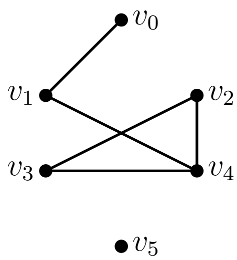
(b) $W_2 = (v_0, v_2)(v_2, v_1)$;
$G_2 = G - \{(v_0, v_2)(v_2, v_1)\}$.



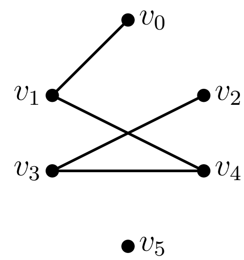
(c) $W_3 = (v_0, v_2)(v_2, v_1)(v_1, v_3)$
$G_3 = G - \{(v_0, v_2)(v_2, v_1)(v_1, v_3)\}$



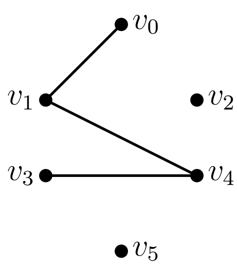
(d) $W_4 = W_3 \to (v_3, v_5)$;
$G_4 = G - E(W_4)$



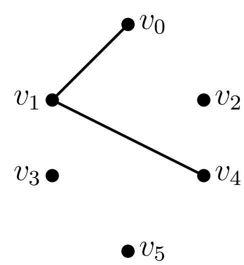
(e) $W_5 = W_4 \to (v_5, v_4)$;
$G_5 = G - E(W_5)$



(f) $W_6 = W_5 \to (v_4, v_2)$;
$G_6 = G - E(W_6)$



(g) $W_7 = W_6 \to (v_2, v_3)$;
$G_7 = G - E(W_7)$



(h) $W_8 = W_7 \to (v_3, v_4)$;
$G_8 = G - E(W_8)$

(i) $W_9 = W_8 \to (v_4, v_1)$; $G_9 = G - E(W_9)$

(j) $W_{10} = W_9 \to (v_1, v_0)$; $G_{10} = G - E(W_{10})$

Figure 5.5: Iterations of Fleury's algorithm.

- Look at the graph G_2 of Figure 5.5b. There are 3 edges incident with v_1. To proceed further, we cannot select the edge (v_1, v_0) (see Step 3 (ii)). We can select (v_1, v_3) or (v_1, v_4). We have decided to select (v_1, v_3) arbitrarily.

Theorem 5.2 (Correctness of the algorithm)**.** *Every trail constructed by Fleury's algorithm is a closed Eulerian trail.*

Proof. Let $W_p = (v_0, e_1, v_2, e_2, \ldots, e_p, v_p)$ be a trail generated by the algorithm.
Claim 1: W_p is a closed trail.

That W_p is a trail is obvious by Step 2, since in every iteration we select an edge which has not been selected in the earlier iterations. Moreover, by (Step 3 (ii)), there are no more edges incident with v_p. Therefore, if $v_0 \neq v_p$, then $deg_G(v_p) = 2k + 1$, where k is the number of times v_p appears internally in W_p, which is a contradiction. Hence, we conclude that $v_0 = v_p$.
Claim 2: W_p contains all the edges of G.

Assume the contrary and let $G_p = G - E(W_p)$. So, $E(G_p) \neq \emptyset$. Hence, there are vertices of positive degree in G_p. Moreover, every vertex in G_p has even degree, since G_p is obtained from G by deleting the edges of a closed trail (see exercise 7). Let

$S = \{v \in V(G_p) : deg_{G_p}(v) > 0\}$. Let $H = [S]$. Then $v_p \in V - S$, since $deg_{G_p}(v_p) = 0$, by Step 3 (i).

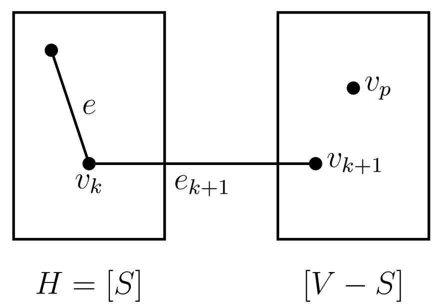


Figure 5.6: A step in the proof of Theorem 5.2.

Let v_k be the last vertex in W_p such that $v_k \in S$. Then $v_{k+1} \in V - S$ and $e_{k+1} = (v_k, v_{k+1})$ is the only edge joining S and $V - S$ in G_p. Therefore we conclude that

(1) e_{k+1} is a cut-edge of G_p.

Next, since every vertex of H has positive degree, there exists an edge e incident with v_k in H. It is not a cut-edge of H, since every vertex in H has even degree (see Exercise 7). e is not a cut-edge of G_p too, since $H \subseteq G_p$. While executing the $(k+1)$-th iteration, we have preferred to select e_{k+1} rather than e. Hence,

(2) e_{k+1} is not a cut-edge of G_p, by Fleury's rule Step 3(ii).

The conclusions (1) and (2) contradict each other. Hence, Claim 2 holds, and W_p is a closed Eulerian trail. $\square$

5.3 An application of Eulerian graphs: Chinese postman problem (Optional)

Problem: As a part of his duties, a postman starts from his office, visits every street at least once, delivers the mail and comes back to the office. Suggest a route of minimum distance.

This optimization problem was first discussed in a paper by Chinese mathematician Mei-Ku Kuan (1962) and hence the problem is named "Chinese Postman Problem".

Graph Theory Model: Given a connected weighted graph $(G, \mathcal{W})$, design an algorithm to find a closed walk of minimum weight containing every edge of G at least once.

It is obvious that if G is Eulerian then one can apply Fleury's algorithm and the resulting closed Eulerian trail is an optimal trail, since every edge appears exactly once. However, if G is non-Eulerian we can construct a super graph G^* which is Eulerian by duplicating certain edges. Note that by duplicating every edge of G, we get a super-graph of G which is Eulerian. So, the problem is to find an optimal set of edges in G whose duplication yields an Eulerian graph. The following algorithm is a solution to Chinese postman problem.

Algorithm:

Input: A connected weighted graph G.

Output: An optimal closed walk of G, containing every edge at least once.

Steps:

(1) If G is Eulerian, apply Fleury's algorithm.

(2) If G is not Eulerian, then identify all the vertices of odd degree, say $v_1, v_2, \ldots, v_{2k}$.

(3) Find a shortest (v_i, v_j)-path P_{ij} for every pair of vertices v_i and v_j by applying Dijkstra's algorithm or Floyd-Warshall algorithm. Let the weight of P_{ij} be $\mathcal{W}_{ij}$.

(4) Construct a complete graph G^* on vertices $z_1, z_2, \ldots, z_{2k}$ $(v_i \leftrightarrow z_i)$ by joining z_i and z_j with an edge of weight $\mathcal{W}_{ij}$.

(5) (Matching problem) Find a set M of k edges say $\{(z_1, z_1^{'}), \ldots, (z_k, z_k^{'})\}$ in G^* such that

(i) no two edges are adjacent;

(ii) subject to (i), M has the minimum weight among all such sets of edges.

(6) In G, duplicate the edges of P_{ij} joining v_i and v_j if $(z_i, z_j) \in M$, to obtain an Eulerian super-graph G^e of G.

(7) Apply Fleury's algorithm to G^e. The resultant closed Eulerian trail is an optimal closed walk of G.

An illustration: Consider the road map G shown in Figure 5.7 with post office located at v_1.

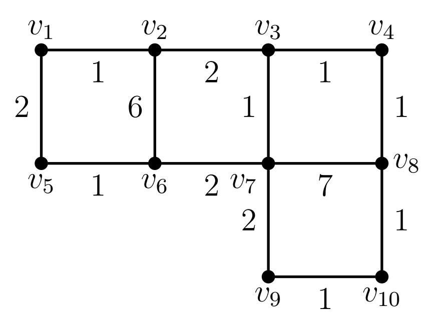


Figure 5.7: An input graph for illustration.

We apply the above seven steps to find an optimal walk containing each edge at least once.

(1) G is not Eulerian.

(2) v_2, v_3, v_6, v_8 are the vertices of odd degrees.

(3) (a) A shortest (v_2, v_3)-path is (v_2, v_3); its weight is 2.
(b) A shortest (v_2, v_6)-path is (v_2, v_1, v_5, v_6); its weight is 4.
(c) A shortest (v_2, v_8)-path is (v_2, v_3, v_4, v_8); its weight is 4.
(d) A shortest (v_3, v_6)-path is (v_3, v_7, v_6); its weight is 3.
(e) A shortest (v_3, v_8)-path is (v_3, v_4, v_8); its weight is 2.
(f) A shortest (v_6, v_8)-path is $(v_6, v_7, v_3, v_4, v_8)$; its weight is 5.

(4) The complete weighted graph G^* constructed by following the steps (3) and (4) is shown in Figure 5.8.

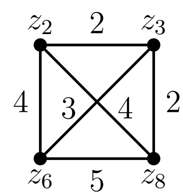


Figure 5.8: A complete weighted graph G^* .

(5) The following are the three sets of two non-adjacent edges in G*.

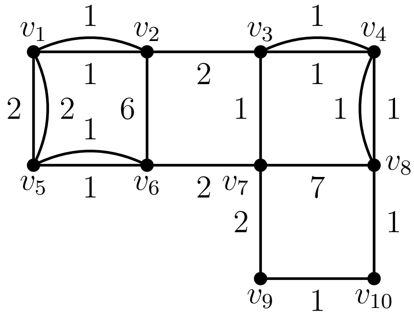
(a) $M_1 = \{(z_2, z_3), (z_6, z_8)\}$; its weight is 7.
(b) $M_2 = \{(z_2, z_6), (z_3, z_8)\}$; its weight is 6.
(c) $M_3 = \{(z_2, z_8), (z_3, z_6)\}$; its weight is 7.

M_2 has the minimum weight.

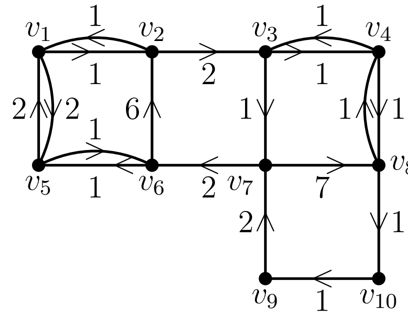
(6) We duplicate the edges of paths $P(v_2, v_6)$ and $P(v_3, v_8)$ in G and obtain the Eulerian graph G^e shown in Figure 5.9a.

(7) We apply Fleury's algorithm to obtain the following optimal closed Eulerian trail, which is a solution to Chinese Postman Problem.

(v_1, v_2) (v_2, v_3) (v_3, v_4) (v_4, v_8) (v_8, v_4) (v_4, v_3) (v_3, v_7) (v_7, v_8) (v_8, v_{10}) (v_{10}, v_9) (v_9, v_7) (v_7, v_6) (v_6, v_5) (v_5, v_1) (v_1, v_5) (v_5, v_6) (v_6, v_2) (v_2, v_1).



(a) G^e



(b) $W(v_1, v_1)$

Figure 5.9: The super-graph G^e of G constructed by following Step (6) and an Eulerian trail $W(v_1, v_1)$.

Exercises

1. Show that a connected graph G is Eulerian iff there exists a partition $(E_1, \ldots, E_k)$ of $E(G)$ such that each $[E_i]$ is a cycle.

2. (a) If in a graph G, every vertex has even degree, then show that G contains no cut-edges.

   (b) If G is Eulerian, show that $k_1(G)$ is even.

3. Draw a simple Eulerian graph G for some n $(4 \leq n \leq 10)$ with $k_0(G) = 1$ and $k_1(G) = 4$.

4. Draw a simple graph with $n = 7$, $\delta(G) \geq 3$ and containing no closed Eulerian trail but containing an open Eulerian trail.

5. Does there exists a simple Eulerian graph on even number of vertices and odd number of edges? Justify your answer.

6. Find the minimum number of edge disjoint trails together containing all the edges of G shown in Figure 5.10.

7. If W is a closed trail in a graph G, then show that $deg_G(x) \equiv deg_H(x) \pmod 2$, for every $x \in V(G)$, where $H = G - E(W)$.

8. Prove or disprove:

   (a) If G is Eulerian, then $L(G)$ is Eulerian.

   (b) If $L(G)$ is Eulerian, then G is Eulerian.

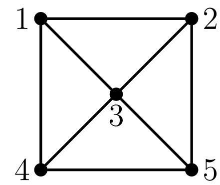


Figure 5.10: A graph G

9. If G is a connected graph, then show that $L(G)$ is Eulerian iff either every vertex in G has even degree or every vertex has odd degree.

10. Given a simple graph G, the p^{th} iterated line graph $L^p(G)$ is defined recursively as follows.
    (i) $L^1(G) = L(G)$,
    (ii) $L^p(G) = L(L^{p-1}(G); p \geq 2$.
    (a) If G a is connected graph $(n \geq 5)$, then show that $L^3(G)$ is Eulerian implies $L^2(G)$ is Eulerian.
    (b) Prove or disprove: $L^2(G)$ is Eulerian $\Rightarrow L(G)$ is Eulerian.

11. If T is a tree, find the minimum number of edges to be added to T to obtain a spanning Eulerian supergraph G^e of T.

12. In the Fleury's algorithm, a graph G^e is constructed by adding the paths P_{ij} in Step (6). Show that G^e is Eulerian.

13. (a) Find conditions on r and s, for $K_r \square K_s$ to be Eulerian.
    (b) Find necessary and sufficient conditions that G and H should satisfy for $G \square H$ to be Eulerian.

14. Let k, n, p be integers ≥ 3. Find necessary and sufficient conditions for $(C_k + K_n) + K_p^c$ to be Eulerian.

15. (a) For which integers $a, b, c, d (\geq 1)$, $K_{a,b,c,d}$ is Eulerian.
    (b) For which values of n, $K_{1,3,5,\ldots,2n-1}$ is Eulerian? Justify.

16. Let G be a connected simple r-regular graph on even number of vertices such that its complement G^c is also connected. Prove or disprove: G or G^c is Eulerian.

17. Show that every connected graph contains a closed walk which contains every edge such that any edge appears at most twice.

18. Solve the Chinese postman problem for the street network shown in Figure 5.11.

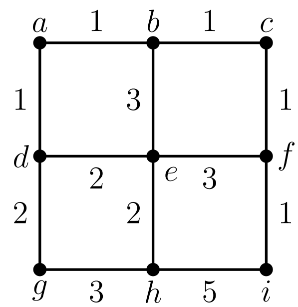


Figure 5.11: A street network.

19. A road map is shown in Figure 5.12. The central strip along each of these roads is to be painted white to have a smooth two-way traffic. The first coordinate in the ordered pair shown along an edge denotes the time for traveling and painting, and the second coordinate denotes the time for traveling without painting. Describe a shortest route if one starts painting from A and comes back to A after finishing the job. What is the total time required for such a route.

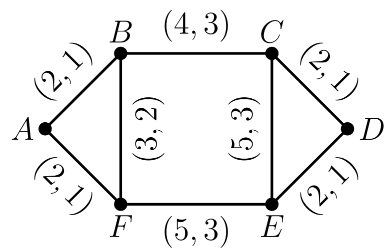


Figure 5.12: A road map.

Modules

Module 6 Hamilton Graphs

Contents

6.1 Introduction

Much research on Hamilton graphs is driven by a real world optimization problem called the ***traveling salesman problem***. It has resemblance to Chinese Postman Problem, but is known to be a hard open problem.

Traveling Salesman Problem (TSP)

Given n cities and distance between any two cities, a traveling salesman wishes to start from one of these cities, visits each city exactly once and comes back to the starting point. Design an algorithm to find a shortest route.

It can be modeled as a problem in several branches of mathematics. Its graph theoretical model is easy to state.

Graph theoretical model of TSP

Given a weighted graph G on n vertices, design an algorithm to find a cycle with minimum weight containing all the vertices of G.

Its solution carries a one million dollar reward by Clay Institute. The unweighted version of the above problem is equally difficult. It is well-known as Hamilton problem in honor of Sir William Rowan Hamilton who used spanning cycles of graphs to construct non-commutative algebras (1859).

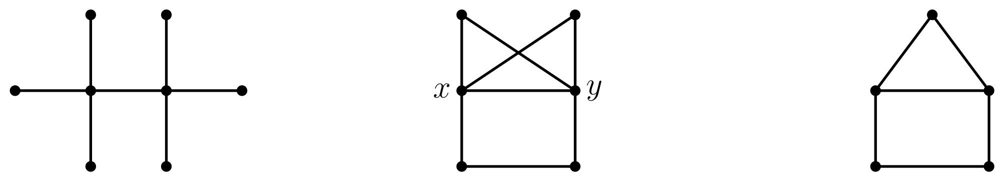
Definitions.

(1) A path (cycle) in a graph containing all its vertices is called a ***Hamilton path*** *(respectively* ***Hamilton cycle****).*

(2) A graph is called a ***Hamilton graph*** *if it contains a Hamilton cycle.*

Remarks.

- Every Hamilton graph is 2-connected (follows by Theorem 2.17).



(a) A graph with no Hamilton path

(b) A graph with a Hamilton path but no Hamilton cycle

(c) A Hamilton graph

Figure 6.1: Examples of Hamilton and non-Hamilton graphs.

- ◦ A graph is Hamilton iff its underlying simple graph is Hamilton. **Hence, the study of Hamilton graphs is limited to simple graphs.**
- ◦ A graph may contain a Hamilton path but may not contain a Hamilton cycle; see Figure 6.1.

Hamilton problems

1. Find necessary and sufficient conditions for a graph to be Hamilton.
2. Design a polynomial-time algorithm to generate a Hamilton cycle in a given graph G or declare that G has no Hamilton cycle.

In the last chapter, we have seen that the above two problems have good solutions with respect to Eulerian graphs. Despite numerous attempts and impressive contributions by various mathematicians and computer scientists, the Hamilton problems have remained open.

6.2 Necessary conditions and sufficient conditions

As remarked earlier all our graphs are simple.

Theorem 6.1 (Necessary condition). *If G is Hamilton, then*

$$c(G - S) \leq |S|, \text{ for every } S \subseteq V(G),$$

where $c(G-S)$ denotes the number of components in $G-S$.

Proof. (Two way counting technique) Let C be a Hamilton cycle in G and let S be a subset of $V(G)$ with s vertices. Consider the set of edges $F \subseteq E(C)$ with one end in S and another end in $G-S$. Every vertex in G is incident with two edges of C.

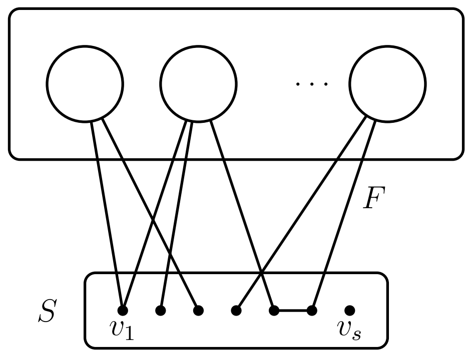


Figure 6.2: Two way counting of $|F|$.

Therefore, every vertex in S is incident with at most two edges of F. Hence,

(1) $|F| \leq 2s = 2|S|$

On the other hand, every component of $G-S$ is incident with at least two edges of F. Hence

(2) $|F| \geq 2c(G-S)$

The inequalities (1) and (2) imply that $c(G-S) \leq |S|$. $\square$

- Using this necessary condition it is easy to show that the graph shown in Figure 6.1b is non-Hamilton; $G - \{x, y\}$ contains 3 components.

Sufficient conditions for the existence of a Hamilton cycle are based on a common intuition that a graph is likely to contain a Hamilton cycle if it contains large number of edges uniformly distributed among the vertices. Many sufficient conditions have been proved by making mathematically precise the term "the large number of edges uniformly distributed among the vertices." We state and prove three such theorems.

Theorem 6.2 (Ore, 1962)**.** *If a graph G on $n(\geq 3)$ is such that*

$$deg(u) + deg(v) \geq n, \text{ for every pair of non-adjacent vertices } u \text{ and } v \tag{Ore}$$

then G is Hamilton.

Proof. (Nash-Williams, 1966). Assume the contrary that G is non-Hamilton though it satisfies (Ore). Let H be a graph obtained from G by successively joining pairs of non-adjacent vertices until addition of any more new edge creates a Hamilton cycle. Then G is a spanning subgraph of H and so $deg_H(v) \geq deg_G(v)$ for every vertex v. Hence, H too satisfies (Ore). Since H is non-Hamilton, there exist two non-adjacent vertices, say u and v. By the construction of H, it follows that $H + (u, v)$ is Hamilton. Hence, there is a Hamilton path $P = (u_1, u_2, \ldots, u_{n-1}, u_n)$ in H connecting u and v, where $u_1 = u$ and $u_n = v$. If u_1 is adjacent to u_k, then u_n is non-adjacent to u_{k-1}.

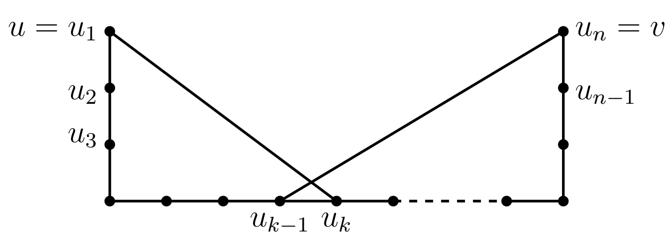


Figure 6.3: Hamilton path P and construction of a Hamilton cycle.

Else,

$$(u_1, u_2, \ldots, u_{k-1}, u_n, u_{n-1}, \ldots, u_k, u_1)$$

is a Hamilton cycle (see Figure 6.3) in H, a contradiction to non-Hamiltonicity of H. Therefore, there are (at least) deg(u) vertices of H which are non-adjacent to v. Hence, $deg(v) \leq (n-1) - deg(u)$, a contradiction to (Ore).

$\square$

Next result due to Dirac (1959) in fact preceded the above theorem.

Corollary (Dirac, 1959)**.** *If a graph G on $n (\geq 3)$ vertices is such that*

$$deg(v) \geq \frac{n}{2}, \text{ for every vertex } v \text{ in } G \qquad \text{(Dirac)}$$

then G is Hamilton.

Proof. An easy consequence of Theorem 6.2. $\square$

It is easily seen that Ore's result is a proper generalization of Dirac's result. That is, we can construct graphs which satisfy (Ore) but do not satisfy (Dirac).

Corollary. *If G is a graph with $n \geq 3$ and $m \geq \dfrac{n^2 - 3n + 6}{2}$, then G is Hamilton.*

Proof. Assume the contrary that G is non-Hamilton. By Theorem 6.2, there exists a pair of non-adjacent vertices u and v such that $deg(u) + deg(v) \leq n - 1$. Hence,

$$\begin{aligned} m &\leq |E(G - \{u, v\})| + deg_G(u) + deg_G(v) \\ &\leq \binom{n-2}{2} + n - 1 \\ &= \frac{n^2 - 3n + 4}{2}, \text{a contradiction.} \qquad \square \end{aligned}$$

Corollary. *If a graph G on n vertices is such that*

$$deg(u) + deg(v) \geq n - 1, \text{ for every pair of non-adjacent vertices } u \text{ and } v \text{ in } G \qquad \text{(P)}$$

then G contains a Hamilton path.

Proof. If $n = 1$, the result is vacuously true. If $n \geq 2$, construct a new graph H by adding a new vertex x and joining x to every vertex of G. Since G satisfies (P), $deg_H(u) + deg_H(v) \geq n+1 = n(H)$, for every pair of non-adjacent vertices u and v in H. So by Theorem 6.2, H contains a Hamilton cycle C. We can choose C so that x is its initial and terminal vertex. So let $C = (x, v_1, v_2, \ldots, v_n, x)$. Then $(v_1, v_2, \ldots, v_n)$ is a Hamilton path in G. $\square$

Theorem 6.3 (Chvatal, 1972)**.** *Let $n \geq 3$. Suppose a graph G with degree sequence $(d_1 \leq d_2 \leq \cdots \leq d_n)$ satisfies the following condition:*

$$\textit{if there is an integer } k \textit{ such that } 1 \leq k < \frac{n}{2} \textit{ and } d_k \leq k, \textit{ then } d_{n-k} \geq n-k \quad \text{(Chvatal).}$$

Then G is Hamilton.

(Essentially, the condition says that if G contains vertices of small degree, then it contains vertices of large degree too. Before proving the theorem, the reader is encouraged to show that if G satisfies (Chvatal), then G contains no vertex of degree 1; see also Exercise 19.)

Proof. (Proof is similar to the proof of Theorem 6.2 but more deeper.) Assume the contrary that G is non-Hamilton though it satisfies (Chvatal). Let H be a graph obtained from G by successively joining pairs of non-adjacent vertices until addition of any more new edge creates a Hamilton cycle. Then G is a spanning subgraph of H and so $deg_H(v) \geq deg_G(v)$ for every vertex v. Hence, H too satisfies (Chvatal). Henceforth, all our statements are with respect to H;in particular, $(d_1 \leq d_2 \leq \cdots \leq d_n)$ is the degree sequence of H. Since H is non-Hamilton, there exist two non-adjacent vertices, say u and v. Among all such pairs of vertices, we choose two non-adjacent vertices u and v such that

(1) $d(u) + d(v)$ is maximum.

Since, $H + (u, v)$ contains a Hamilton cycle, H contains a Hamilton path $P = (u_1, u_2, \ldots, u_{n-1}, u_n)$ connecting u and v, where $u_1 = u$ and $u_n = v$. If u is adjacent to u_j, then v is non-adjacent to u_{j-1}; else, H contains a Hamilton cycle as in the proof of Theorem 6.2. Hence there are at least $d(u)$ vertices which are non-adjacent to v. So,

(2) $d(u) + d(v) \leq n - 1$.

Without loss of generality, assume that

(3) $d(u) \leq d(v)$; and let $d(u) = k$.

So,

(4) $k < \frac{n}{2}$.

By the maximality of $d(u) + d(v)$, every vertex u_{j-1} that is non-adjacent to v has degree at most $d(u)(= k)$. So, there are k vertices of degree at most k. Since we have arranged the degree sequence in non-decreasing order, it follows that

(5) $d_k \leq k$.

Since, u is adjacent to k vertices, it is non-adjacent to $n - 1 - k$ vertices (other than u). Again by the maximality of $d(u) + d(v)$, each of these $n - 1 - k$ vertices has degree at most $d(v)(\leq n - 1 - k)$. Moreover, $d(u) \underset{(3)}{\leq} d(v) \underset{(2)}{\leq} n - 1 - k$. Hence, there are at least $n - k$ vertices of degree at most $n - 1 - k$. Since we have arranged the degree sequence in non-decreasing order, it follows that

(6) $d_{n-k} \leq n - 1 - k$.

Thus we have found a k such that $k \underset{(4)}{<} \frac{n}{2}$, $d_k \underset{(5)}{\leq} k$ and $d_{n-k} \underset{(6)}{\leq} n-1-k$. But the existence of such a k is a contradiction to (Chvatal). $\square$

Remark. It can be shown that if G satisfies (Ore), then it satisfies (Chvatal). Moreover, there exist graphs which satisfy (Chvatal) but do not satisfy (Ore). So Chvatal's result is a proper generalization of Ore's result. For example, consider the graph G shown in Figure 6.4. It does not satisfy (Ore). However, its degree sequence $(2,3,3,4,5,5,5,5)$ satisfies (Chvatal). So, Ore's theorem does not guarantee an Hamilton cycle in G but Chvatal's theorem guarantees a Hamilton cycle.

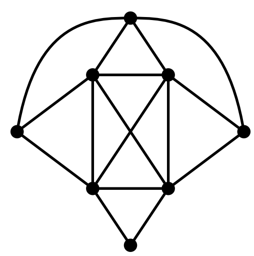


Figure 6.4: A graph G.

For the next theorem, we require a new concept.

Definitions.

- *A set $I \subseteq V(G)$ is called an* ***independent set*** *if no two vertices in I are adjacent.*
- *The parameter $\alpha_0(G) = \max\{|I| : I \text{ is an independent set of } G\}$ is called the* ***independence number*** *of G.*

Clearly, $\alpha_0(P_n) = \lceil \frac{n}{2} \rceil$, $\alpha_0(C_n) = \lfloor \frac{n}{2} \rfloor$, $\alpha_0(K_n) = 1$ and $\alpha_0(K_{m,n}) = \max\{m,n\}$. The independent sets and independence number of a graph will be studied in more detail in Chapter 7.

Theorem 6.4 (Chvatal and Erdös, 1972)**.** *If G is a graph ($n \geq 3$) with*

$$k_0(G) \geq \alpha_0(G), \text{ where } k_0(G) \text{ is the vertex connectivity of } G, \tag{CE}$$

then G is Hamilton.

Proof. (Contradiction method) Assume the contrary that G is non-Hamilton, though it satisfies (CE). Let C be a longest cycle in G; see Figure 6.5. C does not contain all the vertices, since G is non-Hamilton. Therefore, there exists a component B in $G - V(C)$. Assume that the vertices of C are numbered in a clockwise direction. Let x^+ denote the vertex which succeeds the vertex x on C. Define

$$S = \{x \in V(C) : x \text{ is adjacent to a vertex in } B\}$$

and

$$S^+ = \{x^+ \in V(C) : x \in S\}.$$

Claim 1: If $x \in S$, then $x^+ \notin S$.

On the contrary, let b_1 and b_2 be the neighbors of x and x^+ (respectively) in B. Let $P(b_1, b_2)$ be a path in B. Then the cycle

$$(x, b_1, P(b_1, b_2), b_2, x^+, C(x^+, x))$$

contains more number of vertices than C, a contradiction; see Figure 6.5.

It follows that S is a vertex-cut; so $k_0(G) \leq |S|$.

Claim 2: S^+ is an independent set in G.

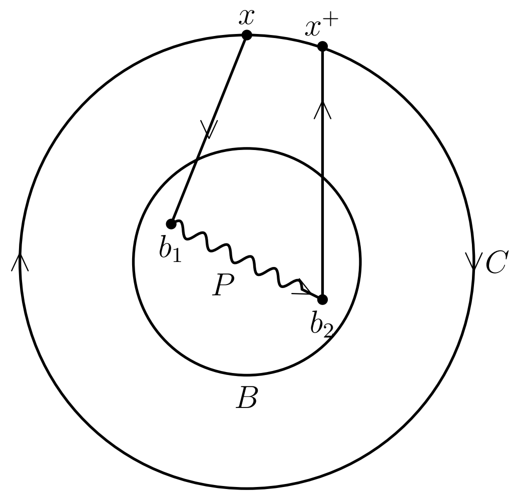


Figure 6.5: Pictorial description for the proof of Claim 1.

On the contrary, suppose that $x^+, y^+ \in S^+$ are adjacent. Let b_1 and b_2 be the vertices of B adjacent to x and y, respectively and let $P(b_1, b_2)$ be a path in B; see Figure 6.6. Then the cycle

$$(x, b_1, P(b_1, b_2), b_2, y, C(y, x^+), y^+, C(y^+, x))$$

contains more number of vertices than C, a contradiction.

Using claims 1 and 2, we conclude that if $b \in B$, then $S^+ \cup \{b\}$ is also an independent set in G. Hence, $k_0(G) \leq |S| = |S^+| < \alpha_0(G)$, a contradiction. So C is a Hamilton cycle. $\square$

Exercises

(All graphs are simple)

1. Draw the following graphs.
   (a) Hamilton and Eulerian.

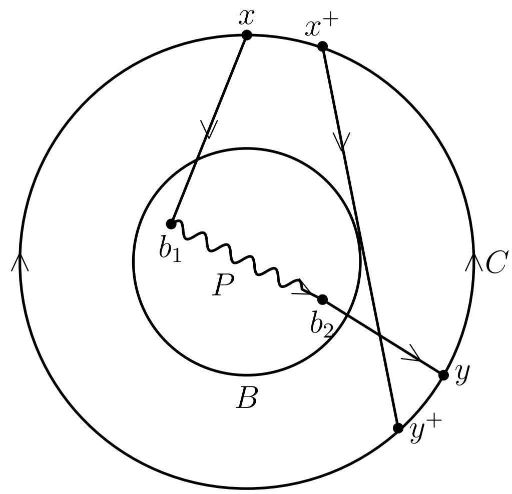


Figure 6.6: Pictorial description for the proof of Claim 2.

(b) Hamilton and non-Eulerian.

(c) non-Hamilton and Eulerian.

(d) non-Hamilton and non-Eulerian.

2. Show that the d-cube Q_d is Hamilton for every $d \geq 2$.

3. (a) Show that the Petersen graph is non-Hamilton.

   (b) Prove or disprove: If P is the Petersen graph, then $P - v$ is Hamilton for every $v \in V(P)$.

4. If G is a non-Hamilton graph such that $G - v$ is Hamilton for every vertex $v \in V(G)$, then show that

   (a) $deg(v) \geq 3$, for every $v \in V(G)$.

   (b) $deg(v) \leq \frac{n-1}{2}$, for every $v \in V(G)$.

   (c) $G - u - v$ is connected for every subset $\{u, v\}$ of $V(G)$.

   (d) $n \geq 10$.

   (e) If $n = 10$, then G is 3-regular.

5. Find 5 edge-disjoint Hamilton cycles in K_{11}.

6. If a graph G has a Hamilton path, then show that

$$c(G - S) \leq |S| + 1, \text{ for every proper subset of } V(G).$$

7. If $deg(u) + deg(v) \geq n+2$, for every pair of vertices u and v in a graph G, then show that G contains two edge-disjoint Hamilton cycles.

8. Prove that the line graph $L(G)$ of a graph G is Hamilton iff G contains a closed trail W such that every edge of G is incident with a vertex of W.

9. Draw the following graphs :
   (a) A 2-connected non-Hamilton graph on at most 8 vertices.
   (b) A non-Hamilton graph on at most 10 vertices with minimum degree at least 3 which has a Hamilton path.
   (c) A graph on at least 9 vertices such that both G and its complement G^c are connected but both are non-Hamilton.

10. For every k, construct a k-connected simple non-Hamilton graph.

11. Draw a non-Hamilton simple graph with 10 vertices such that for every pair u, v of non-adjacent vertices, we have $deg(u) + deg(v) \geq 9$.

12. Draw the following non-Hamilton graphs.
   (a) $m = \frac{n^2-3n+4}{2}$.
   (b) $deg(u) + deg(v) \geq n-1$, for every pair of non-adjacent vertices in G.
   (c) $\delta(G) \geq \frac{n-1}{2}$.

13. Let (v_1, v_2, v_3, v_4) be a path P_4 on 4 vertices. Show that the composition graph $P_4(K_n^c, K_n, K_n, K_n^c)$ on $4n$ vertices is a self-complimentary non-Hamilton graph.

14. (a) For what values of p, $K_{1,2,\ldots,p}$ is Hamilton? Justify your answer.
   (b) For what integers n, the graph $K_{2,3,4,n}$ is Hamilton? Justify your answer.
   (c) Show that a complete r-partite graph $K_{n_1,n_2,\ldots,n_r}$ $(n_1 \leq n_2 \leq \cdots \leq n_r)$ is Hamilton iff $n_1 + n_2 + \cdots + n_{r-1} \geq n_r$.
   (d) Show that $K_s + K_{1,t}$ is Hamilton if and only if $t \leq s+1$.

15. Find necessary and sufficient conditions for the following graphs to be Hamilton.
   (a) $P_s \Box P_t$.
   (b) $K_s^c + K_t$.

16. Let G be a graph obtained from K_n $(n \geq 3)$ by deleting any set of at most $n-3$ edges. Using Ore's theorem or otherwise show that G is Hamilton.

17. Show that if a graph satisfies (Ore), then it satisfies (Chvatal).

18. Give an example of a graph which satisfies (Chvatal) but does not satisfy (Ore).

19. If G satisfies (Chvatal) and $d_2 \leq 2$, then show that G contains at least 3 vertices of degree $\geq n-3$.

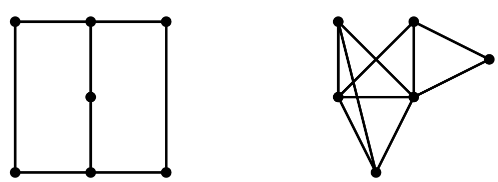
20. Which of the following degree sequences of graphs satisfy (Chvatal)(see Theorem 6.3).
    (a) (2,2,2,2,2)
    (b) (2,2,3,3,3,3)
    (c) (3,3,3,3,3,3)
    (d) (2,2,3,4,4,5)

21. Which of the following conditions are satisfied by the graph $K_3 + (K_3^c \cup K_2)$?
    (a) Ore's sufficient condition for a graph to be Hamilton.
    (b) Chvatal's sufficient condition for a graph to be Hamilton.
    (c) Chvatal's necessary condition for a graph to be Hamilton.

22. Let G_1 and G_2 be two simple graphs both on n vertices and satisfying Dirac's condition for the existence of a Hamilton cycle. Let G be a simple graph obtained from G_1 and G_2 by joining edges between G_1 and G_2 such that:
    (a) every vertex in G_1 is joined to at least half the number of vertices in G_2, and
    (b) every vertex in G_2 is joined to at least half the number of vertices in G_1.

    Show that G is Hamilton.

23. Prove or disprove : Graphs shown in Figure 6.7 are Hamilton.



(a) (b)

Figure 6.7

The ***closure*** $C(G)$ of a graph G is the graph obtained from G by successively joining pairs of non-adjacent vertices whose degree sum is at least n, until no such pair remains. For example, see Figure 6.8.

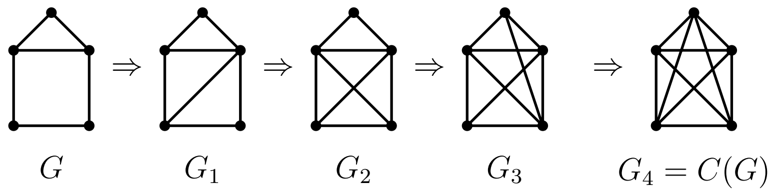


Figure 6.8: Construction of closure of a graph.

---

24. (a) Find the closures of the graphs shown in Figure 6.9.

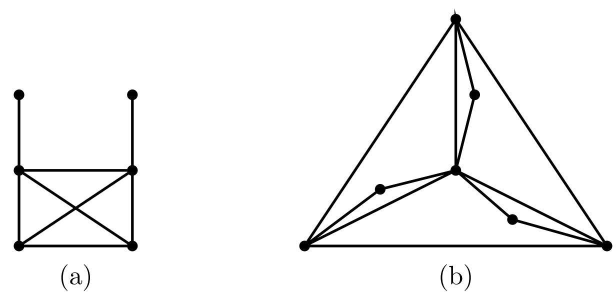


Figure 6.9

(b) Draw an example of a graph G whose closure is neither complete nor G.

25. (a) Let u, v be two non-adjacent vertices in G such that $deg(u) + deg(v) \geq n$. Show that G is Hamilton iff $G + (u, v)$ is Hamilton.

(b) Show that G is Hamilton iff $C(G)$ is Hamilton.

(c) If G satisfies (Chvatal)(Theorem 6.3), then show that $C(G)$ is complete.

A graph G is said to be ***Hamilton-connected***, if any two vertices in G are connected by a Hamilton path.

---

26. (a) Show that $K_5 - e$, where e is an edge, is Hamilton-connected.

(b) Show that $K_4 - e$, where e is an edge, is not Hamilton-connected.

(c) If G is Hamilton-connected, then show that every edge in G belongs to a Hamilton cycle.

(d) Let G be a graph and let w be a vertex of degree 2 adjacent with vertices u and v. Let H be the graph $G-w$ if $(u, v) \in E(G)$; $G-w+(u, v)$, otherwise. Show that G is Hamilton-connected iff H is Hamilton-connected.

(e) If G is Hamilton-connected, then show that $m \geq \lfloor \frac{3n+1}{2} \rfloor$, for $n \geq 4$.

(f) If G is a graph with $m \geq \frac{(n-1)(n-2)}{2} + 3$, then show that G is Hamilton-connected.

Module 7 Independent sets, coverings and matchings

Contents

7.1 Introduction

The concepts of independent sets and coverings are used in modeling several real world problems. One of the real world problems called the *Job Assignment Problem* has motivated much research on matchings.

Job-Assignment-Problem (JAP)

There are s persons and t jobs. Each person is capable of handling certain jobs. Under what conditions we can employ **each** of the p persons with a job he/she is capable of handling? The rule of **one-person-one-job** is assumed.

- Notice that some jobs may be left undone.
- Model the problem using appropriate graph theoretic terminology.
- Another popular version of the JAP is known as ***The Marriage Problem***. There are s girls and t boys. Each girl has a list of favorite boys. Under what conditions one can marry off each of the girls to a boy in her list?
- A pure set-theoretic version of JAP is known as ***The Problem of Set Representatives***. Given a set X and a family $F = (X_1, X_2, \ldots, X_s)$ of subsets of X, find necessary and sufficient conditions for choosing s distinct elements $(x_1, x_2, \ldots, x_s)$, such that $x_i \in X_i$, $i = 1, 2, \ldots, s$. The element x_i is called the ***representative*** of X_i.

Notice that if X, F, X_i and s are infinite, then it is a variation of "Axiom of Choice".

7.2 Independent sets and coverings: basic equations

We start with a series of definitions.

Definitions. *Let G be a graph.*

- *A set of vertices I is called an* ***independent set*** *if no two vertices in I are adjacent. An independent set is also called a* ***stable set****.*

  - Any singleton set is an independent set. So one is interested to find a largest independent set.

- *The parameter* $\alpha_0(G) = \max\{|I| : I \text{ is an independent set in } G\}$ *is called the* ***vertex-independence number*** *of* G.

- *Any independent set* I *with* $|I| = \alpha_0(G)$ *is called a* ***maximum independent set***.

- *An independent set* I *is called a* ***maximal independent set*** *if there is no independent set which properly contains* I.

  - Clearly, any maximum independent set is a maximal independent set but a maximal independent set need not be a maximum independent set. An illustration is given below.

- *If* $e(u, v)$ *is an edge in* G*, then* e *is said to* ***cover*** u *and* v *and vice versa.*

- *A set of vertices* K *is called a* ***vertex-cover*** *of* G *if every edge in* G *is covered by a vertex in* K*. That is, if every edge in* G *has at least one of its end vertices in* K.

  - Clearly, $V(G)$ is a vertex-cover of G. So one is interested to find a smallest vertex cover.

- *The parameter* $\beta_0(G) = min\{|K| : K \text{ is a vertex cover of } G\}$ *is called the* ***vertex-covering number*** *of* G.

- *Any vertex-cover* K *with* $|K| = \beta_0(G)$ *is called a* ***minimum vertex-cover***.

- *A vertex-cover* K *is called a* ***minimal vertex-cover*** *if there is no vertex-cover which is properly contained in* K.

  - Clearly, every minimum vertex-cover is a minimal vertex-cover but a minimal vertex-cover need not be a minimum vertex-cover. An illustration is given below.

An illustration: In the graph shown in Figure 7.1, $\{1, 2, 4\}$, $\{3, 5, 6\}$ are maximal independent sets, but they are not maximum independent sets; $\{1, 2, 5, 6\}$ is the

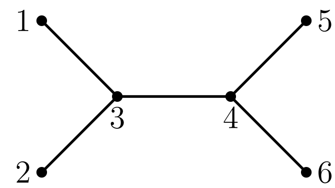


Figure 7.1: An example to illustrate the independent sets and vertex covers.

maximum independent set. The vertex subset $\{1, 2, 4\}$ is a minimal vertex-cover but it is not a minimum vertex-cover; {3,4} is a minimum vertex-cover.

This example illustrates that a graph may contain many maximal and maximum independent sets. Similarly, it may contain many minimal and minimum vertex-covers. However, $\alpha_0(G)$ and $\beta_0(G)$ are unique.

The following table shows the independence number and vertex-covering number of some standard graphs.

	P_n	C_n	K_n	K_n^c	$K_{r,s}$
α_0	$\lceil\frac{n}{2}\rceil$	$\lfloor\frac{n}{2}\rfloor$	1	n	$\max\{r, s\}$
β_0	$\lfloor\frac{n}{2}\rfloor$	$\lceil\frac{n}{2}\rceil$	$n-1$	0	$\min\{r, s\}$

Table 7.1: Table of independence numbers and covering numbers.

The reader may notice that for every graph G shown in the above table, $\alpha_0(G) + \beta_0(G) = |V(G)|$. In fact, this equation holds for any arbitrary graph G. Before proving this claim, we observe a stronger statement.

Theorem 7.1. *I is an independent set in G iff $V(G) - I$ is a vertex-cover of G.*

Proof. I is an independent set

$\Leftrightarrow$ no two vertices in I are adjacent

$\Leftrightarrow$ no edge has both of its end vertices in I

$\Leftrightarrow$ every edge in G has an end vertex in $V(G) - I$

$\Leftrightarrow$ $V(G) - I$ is a vertex-cover of G. $\square$

Corollary. *For any graph G, $\alpha_0(G) + \beta_0(G) = n(G)$.*

Proof. Let I be a maximum independent set. By the above theorem, $V(G) - I$ is a vertex cover of G. Hence, $\beta_0(G) \leq |V(G)-I| = n-\alpha_0(G)$; that is $\alpha_0(G)+\beta_0(G) \leq n$.

Let K be a minimum vertex-cover of G. By the above theorem, $V(G) - K$ is an independent set of vertices. Hence, $\alpha_0(G) \geq |V(G) - K| = n(G) - \beta_0(G)$; that is $\alpha_0(G) + \beta_0(G) \geq n(G)$.

The two inequalities imply the corollary. $\square$

We now define the edge analogues of independent sets of vertices and vertex-covers.

Definitions.

- *A subset of edges M in G is called an **independent set** of edges if no two edges in M are adjacent. An independent set of edges is more often called as a **matching.***
  - Clearly, if M is a singleton, then it is a matching. So one is interested to find a largest matching.
- *The parameter $\alpha_1(G) = max\{|M| : M \text{ is a matching in } G\}$ is called the **matching number** of G.*
- *Any matching M with $|M| = \alpha_1(G)$ is called a **maximum matching**.*
- *A matching M is called a **maximal matching** if there is no matching which properly contains M.*

- *A set of edges F is called an* ***edge-cover*** *of G if every vertex in G is incident with an edge in G.*
  - A graph G need not have an edge cover; for example, $K_2 \cup K_1$ has no edge-cover. In fact, a graph G has an edge cover iff $\delta(G) > 0$.
  - If $\delta(G) > 0$, then $E(G)$ is an edge-cover. So one is interested to find a smallest edge-cover.
- *If $\delta(G) > 0$, then the parameter $\beta_1(G) = min\{|F| : F$ is an edge cover of $G\}$ is called the* ***edge covering number*** *of G.*
- *Any edge-cover F with $|F| = \beta_1(G)$ is called a* ***minimum edge cover.***
- *An edge-cover F is called a* ***minimal edge-cover*** *if there is no edge-cover which is properly contained in F.*

An illustration: In the graph of Figure 7.2, $\{a, d\}$ and $\{b, c\}$ are maximal matchings but they are not maximum matchings ; $\{a, e, f\}$ is a maximum matching. $\{b, c, e, f\}$ is a minimal edge-cover but it is not a minimum edge-cover. $\{a, e, f\}$ is a minimum edge-cover.

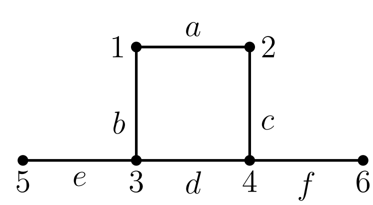


Figure 7.2: An example to illustrate matchings and edge covers.

Again we observe that a graph G may contain many maximal and maximum matchings. It also may contain many minimal and minimum edge covers. But $\alpha_1(G)$ and $\beta_1(G)$ are unique.

The following table shows the matching-number and edge-covering number of some standard graphs.

	P_n	C_n	K_n	$K_{r,s}$
α_1	$\lfloor \frac{n}{2} \rfloor$	$\lfloor \frac{n}{2} \rfloor$	$\lfloor \frac{n}{2} \rfloor$	$\min\{r, s\}$
β_1	$\lceil \frac{n}{2} \rceil$	$\lceil \frac{n}{2} \rceil$	$\lceil \frac{n}{2} \rceil$	$\max\{r, s\}$

Table 7.2: Matching numbers and edge-covering numbers.

Again notice that for every graph G shown in the above table $\alpha_1(G)+\beta_1(G) = n(G)$. However, the edge analogue of Theorem 7.1 does not hold. That is, if M is a matching in G, then $E(G) - M$ need not be an edge cover of G. And if F is an edge cover of G, then $E(G) - F$ need not be a matching. Reader is encouraged to construct graphs to justify these claims. However, $\alpha_1(G) + \beta_1(G) = n(G)$, for any graph G with $\delta(G) > 0$. Before proving it we introduce a new concept.

Definitions.

- *Let M be a matching in G. A vertex v is said to be **M-saturated** if there exists an edge in M which is incident to v; else v is said to be **M-unsaturated.***
- *A matching M is said to be a **perfect matching** if it saturates every vertex of G.*

In the following graph, $M_1 = \{(1, 2), (4, 5)\}$ is a matching. The vertices 1, 2, 4 and 5 are M_1-saturated but 3 and 6 are M_1-unsaturated. So, it is not a perfect matching. However, $M_2 = \{(1, 2), (3, 4), (5, 6)\}$ is a perfect matching.

Remarks.

- Every perfect matching is a maximum matching. However, the converse is false.
- If G has perfect matching, then $n(G)$ is even. However, the converse is false; for example, $K_{1,3}$ and $K_{2,4}$ do not have perfect matchings.

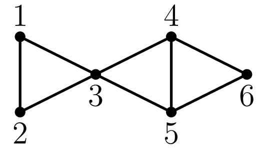


Figure 7.3: A graph to illustrate perfect matching.

Theorem 7.2. *For any graph G with $\delta(G) > 0$, $\alpha_1(G) + \beta_1(G) = n(G)$.*

Proof. It consists of two steps.

(1) $\alpha_1(G) + \beta_1(G) \leq n(G)$.

Let M be a maximum matching and U be the set of all M-unsaturated vertices. Then $|U| = n - 2|M|$. Let $U = \{v_1, v_2, \ldots, v_p\}$, where $p = n - 2|M|$. Let e_i be an edge incident to v_i, $i = 1, \ldots, p$; such an edge exists, since $\delta(G) > 0$. Then $M \cup \{e_1, e_2, \ldots, e_p\}$ is an edge-cover. So,

$$\beta_1(G) \leq |M| + p \leq |M| + (n - 2|M|) = n - |M| = n - \alpha_1(G).$$

Hence, $\alpha_1(G) + \beta_1(G) \leq n$.

(2) $\alpha_1(G) + \beta_1(G) \geq n(G)$.

Let F be a minimum edge cover. Let H be the spanning subgraph of G with edge set F. Let M_H be a maximum matching in H and U be the set of M_H-unsaturated vertices in H. As before, $|U| = n - 2|M_H|$. Let $U = \{v_1, v_2, \ldots, v_p\}$, where $p = n - 2|M_H|$. Since, F is an edge-cover, it contains an edge e_i incident with v_i, $1 \leq i \leq p$. Since, M_H is a maximum matching, U is an independent set in H. Therefore $e_i \neq e_j$, if $v_i \neq v_j$. Hence, $e_1, e_2, \ldots, e_p$ are all distinct edges. It follows that $|F| \geq |M_H| + p = |M_H| + n - 2|M_H| = n - |M_H|$. Hence, $\beta_1(G) = |F| \geq n - |M_H| \geq n - \alpha_1(G)$, since $\alpha_1(G) \geq |M_H|$. So, $\alpha_1(G) + \beta_1(G) \geq n$.

The two inequalities imply the theorem. □

7.3 Matchings in bipartite graphs: Hall's theorem and König's theorem

We have now enough terminology to model the job-assignment-problem and solve it.

Graph theoretic model: Let $P_1, P_2, \dots, P_s$ be the s persons and $J_1, J_2, \dots, J_t$ be the t jobs. Represent each P_i by a vertex p_i and each job J_h by vertex j_h. Join p_i and j_h by an edge if P_i is capable of handling j_h. This construction yields a bipartite graph $G[X, Y]$, where $X = \{p_1, \dots, p_s\}$ and $Y = \{j_1, \dots, j_t\}$. The problem is to find necessary and sufficient conditions for G to contain a matching saturating every vertex in X. The following example illustrates the construction of G.

Persons	Jobs P_i can handle
P_1	J_1, J_3
P_2	J_1, J_2, J_3
P_3	J_2, J_4, J_5
P_4	J_3

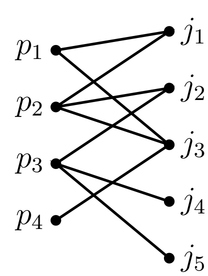


Figure 7.4: A job assignment as required is possible in this example. One possible assignment is to assign P_1 to J_1, P_2 to J_2 P_3 to J_5 and P_4 to J_3.

If G$[X, Y]$ contains a matching, say $\{(p_1, j_1), \dots, (p_s, j_s)\}$ saturating every vertex in X, then p_i can be assigned the job j_i, $1 \le i \le s$.

A necessary condition for the existence of a solution for JAP is intuitively obvious: Every set of k persons $(1 \le k \le s)$ must be able to handle at least k jobs together. That is, in graph theoretic terminology, if G has a matching saturating every vertex in X, then every k vertices in X are adjacent to at least k vertices in Y $(1 \leq k \leq s)$. Hall (1939) showed that the converse is also true.

Notation: If $G[X, Y]$ is a bipartite graph and $S \subseteq X$, let $N_G(S) = \{y \in Y : y \text{ is adjacent to some vertex in } S\}$.

• Hall's Theorem

Theorem 7.3 (Hall, 1939)**.** *A bipartite graph $G[X, Y]$ has a matching saturating every vertex in X iff*

(Hall) $\quad |N_G(S)| \geq |S|$, *for every* $S \subseteq X$.

Proof. Let $X = \{x_1, x_2, \ldots, x_s\}$ and $Y = \{y_1, y_2, \ldots, y_t\}$.

Necessity $(\Rightarrow)$**:** Suppose G has a matching $M = \{(x_1, y_1), \ldots, (x_s, y_s)\}$ saturating every vertex in X. Let $S = \{x_1', x_2', \ldots, x_r'\} \subseteq X$. Then $y_1', y_2', \ldots, y_r' \in N_G(S)$ and they are all distinct, since M is a matching. So $|N_G(S)| \geq r = |S|$.

Sufficiency $(\Leftarrow)$**:** There are many proofs of sufficiency. The following proof is due to Halmos and Vaughan (1950). It is by induction on s.

If $s = 1$, then the result is obvious. So, let $s \geq 2$ and proceed to the next step in the induction. We make two cases.

Case 1: For every proper subset S of X, $|N_G(S)| \geq |S| + 1$.
Since G satisfies (Hall), $|N(\{x_1\})| \geq 1$; and so there is a vertex, say $y_1 \in Y$ such that (x_1, y_1) is an edge. Consider the subgraph $L = G - \{x_1, y_1\}$ whose bipartition is $[X_1, Y_1]$, where $X_1 = \{x_2, \ldots, x_s\}$ and $Y_1 = \{y_2, \ldots, y_t\}$. We claim that $L(X_1, Y_1)$ satisfies (Hall). So, let $T \subseteq X_1$. Clearly, $N_G(T) \subseteq N_L(T) \cup \{y_1\}$. Hence,

$$\begin{aligned} |N_L(T)| &\geq |N_G(T)| - 1, \\ &\geq (|T| + 1) - 1, \text{by our assumption}, \\ &= |T|. \end{aligned}$$

Therefore, by induction hypothesis, $L[X_1, Y_1]$ contains a matching M saturating every vertex in X_1. Then $M \cup \{(x_1, y_1)\}$ is a required matching of G.

Case 2: For some proper subset S of X, $|N_G(S)| = |S|$.

For notational convenience, let $S = \{x_1, x_2, \ldots, x_p\}$ and $N(S) = \{y_1, y_2, \ldots, y_p\}$.

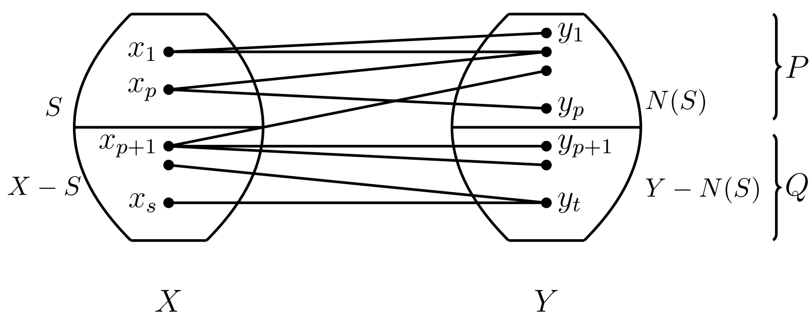


Figure 7.5: The bipartite graphs $P[S, N(S)]$ and $Q[X - S, Y - N(S)]$.

Consider the bipartite subgraphs $P[S, N(S)]$ and $Q[X - S, Y - N(S)]$.

We shall prove that both these graphs satisfy (Hall) and then complete the proof by appealing to the induction hypothesis.

We consider first $P[S, N(S)]$. Let $T \subseteq S$. Clearly, by our assumption of S, $N_P(T) = N_G(T)$; so $|N_P(T)| \geq |T|$. Thus P satisfies (Hall), and hence, by induction hypothesis, P has a matching, say M_1 saturating every vertex in S. Next, consider $Q[X - S, Y - N(S)]$. Let $T \subseteq X - S$. Clearly, $N_G(S \cup T) \subseteq N_G(S) \cup N_Q(T)$. So,

$$\begin{aligned}
|N_Q(T)| &\geq |N_G(S \cup T)| - |N_G(S)|, \\
&\geq |S \cup T| - |S|, \text{ since G satisfies (Hall)}, \\
&= |S| + |T| - |S|, \text{ since } S \text{ and } T \text{ are disjoint}, \\
&= |T|.
\end{aligned}$$

Thus $Q[X - S, Y - N_G(S)]$ satisfies (Hall). So by induction hypothesis, Q has a matching, say M_2 saturating every vertex in $X - S$. Then $M_1 \cup M_2$ is a required matching. $\square$

Two interesting corollaries follow.

Corollary. *Let G be a bipartite graph with bipartition $[X, Y]$ and $\delta(G) > 0$. If*

$$\min_{x \in X} deg(x) \geq \max_{y \in Y} deg(y),$$

then G contains a matching saturating every vertex in X.

Proof. It is enough if we prove that G satisfies (Hall). So, let $S \subseteq X$. Let $\delta(X) = \min_{x \in X} deg(x)$ and $\Delta(Y) = \max_{y \in Y} deg(y)$. If e is an edge incident with a vertex in X, then its other end vertex is in $N(S)$. So, there are at least $|S| \cdot \delta(X)$ edges incident with the vertices of $N(S)$. Hence, there is a vertex $y \in N(S)$ which is incident with at least $\dfrac{|S| \cdot \delta(X)}{|N(S)|}$ edges ($\because$ Use the pigeon-hole principle). So, $\Delta(Y) \geq deg(y) \geq \dfrac{|S| \cdot \delta(X)}{|N(S)|}$. Since, $\delta(X) \geq \Delta(Y)$, we obtain $|N(S)| \geq |S|$. And thus, we have verified (Hall). $\square$

Corollary. *Let $G[X, Y]$ be a k-regular $(k \geq 1)$ bipartite graph. Then $E(G)$ can be partitioned into k sets $E_1, E_2, \ldots, E_k$, where each E_j is a perfect matching.*

Proof. We shall prove the theorem by induction on k. At the outset observe that $|X| = |Y|$, since $k|X| = |E(G)| = k|Y|$. If $k = 1$, then $E(G)$ is a perfect matching and hence the assertion follows. So we proceed to the next step in the induction.

Since, $G[X, Y]$ is a k-regular bipartite graph it satisfies the hypothesis of the above corollary. Hence, it contains a perfect matching, say E_1. Since, $|X| = |Y|$, $G - E_1$ is a $(k - 1)$-regular bipartite graph. Therefore, by induction hypothesis,

$E(G - E_1)$ can be partitioned into $k - 1$ sets, say $E_2, \ldots, E_k$, where each E_i is a perfect matching. Then $(E_1, E_2, \ldots, E_k)$ is a required partition of $E(G)$. $\square$

• König's Theorem

There are several theorems in discrete mathematics which show that a minimum parameter is equal to a maximum parameter. The next result is one such theorem. Before stating it, we observe an important inequality. If M is a matching and K is a vertex-cover of G, then any vertex of K covers at most one edge of M. Hence:

$$\boxed{\beta_0(G) \geq \alpha_1(G), \text{ for any graph } G.}$$

Remarks.

- $\beta_0(C_{2k}) = \alpha_1(C_{2k}) = k$.
- $\beta_0(K_{2p}) = \alpha_1(K_{2p}) = p$.
- $\beta_0(C_{2k+1}) = k + 1$, $\alpha_1(C_{2k+1}) = k$.

The above remarks lead to the following unsolved question.

$$\boxed{\text{For which graphs } G, \ \beta_0(G) = \alpha_1(G)?}$$

The following result identifies one such class of graphs.

Theorem 7.4 (König, 1931)**.** *For any bipartite graph $G[X, Y]$,*

$$\beta_0(G) = \alpha_1(G).$$

Proof. In view of the above observation, we have to only show that $\alpha_1(G) \geq \beta_0(G)$. Let $K = A \cup B$ be a minimum vertex-cover, where $A \subseteq X$ and $B \subseteq Y$. So, $|A| + |B| = |K| = \beta_0(G)$.

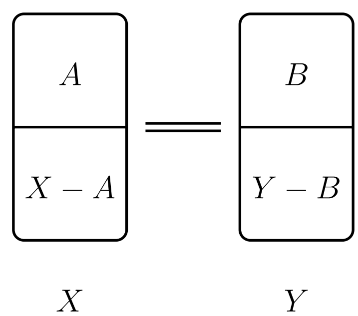


Figure 7.6: A bipartite graph with vertex-cover $A \cup B$.

We consider the bipartite subgraphs $P[A, Y - B]$ and $Q[B, X - A]$.

Claim 1: $P[A, Y - B]$ has a matching M_1 saturating every vertex in A.

To prove this claim it is enough if we verify that P satisfies (Hall). On the contrary, suppose that there exists some $S \subseteq A$ such that $|N_P(S)| < |S|$. Then it is easy to verify that the set $(A - S) \cup N_P(S) \cup B$ is a vertex cover of G. So,

$$\begin{aligned} |(A - S) \cup N_P(S) \cup B| &= |A| - |S| + |N_P(S)| + |B|, \\ &< |A| + |B|, \text{ since by our assumption, } |N_p(S)| < |S|, \\ &= |K|. \end{aligned}$$

This is a contradiction to the minimality of K. Therefore, by Hall's theorem, there exists a matching M_1 in P saturating every vertex in A.

Claim 2: Similarly, it follows that $Q[B, X - A]$ has a matching M_2 saturating every vertex in B.

Clearly $M_1 \cap M_2 = \emptyset$. Hence,

$$\begin{aligned} \alpha_1(G) &\geq |M_1| + |M_2|, \\ &= |A| + |B|, \\ &= |K| = \beta_0(G). \end{aligned}$$

□

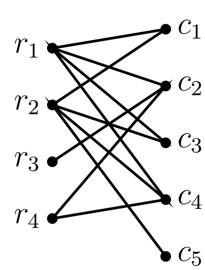
An equivalent form of König's theorem, in matrix terminology, is due to Egerváry (1931). To state it we require new terminology.

Definition. *A **line** of a matrix is either a row or a column. Let $A = [a_{ij}]$ be a $m \times n$ matrix where each a_{ij} is 0 or 1. A set I of $1's$ in A is said to be **independent** if no two $1's$ in I lie on a common line. A line h of A is said to **cover** a 1 if the 1 lies in h.*

An illustration:

$$\begin{array}{cc} & \begin{array}{ccccc} c_1 & c_2\downarrow & c_3 & c_4\downarrow & c_5 \end{array} \\ \begin{array}{c} r_1 \\ \overrightarrow{r_2} \\ r_3 \\ r_4 \end{array} & \begin{pmatrix} 0 & \boxed{1} & 0 & 0 & 0 \\ \boxed{1} & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \boxed{1} & 0 \end{pmatrix} \end{array}$$

A



$G[R, C]$

Figure 7.7: A matrix A with 3 independent $1's$ and 3 lines covering all the $1's$ in A. (b) A bipartite graph $G[R, C]$ representing A, defined in the proof of Theorem 7.5.

Theorem 7.5 (Egerváry, 1931)**.** *Let $A = [a_{ij}]$ be a $m \times n$ matrix, where a_{ij} = 0 or 1. Then, the maximum number of independent $1's$ in A is equal to the minimum number of lines which cover all the 1's in A.*

Proof. Associate a bipartite graph $G[R, C]$ with A as follows; see Figure 7.7.

- Represent each row R_i $(1 \leq i \leq m)$ of A by a vertex r_i.
- Represent each column C_j $(1 \leq i \leq n)$ of A by a vertex c_j.
- Join r_i and c_j iff $a_{ij} = 1$.

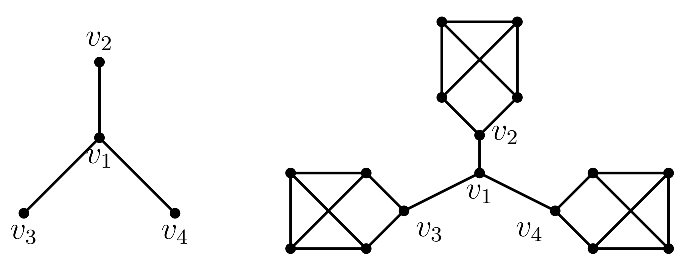
The following assertions can be easily observed.

- There is a 1-1 and onto function between the set of all $1's$ in A and the set of all edges in G, namely $a_{ij} = 1$ iff $(r_i, c_j) \in E(G)$.
- Two $1's$, say $a_{ij} = 1$ and $a_{pq} = 1$ are independent in A iff the corresponding edges (r_i, c_j) and (r_p, c_q) are independent in G. Therefore, I is a set of independent $1's$ in A iff the corresponding set M of edges in G is a matching in G.
- A set of lines covers all the $1's$ in A iff the corresponding set of vertices in G covers all the edges in G.

Hence the result follows. $\square$

7.4 Perfect matchings in graphs

All the graphs in this section are simple graphs. If G has a perfect matching, then $n(G)$ is even, obviously. However, the converse does not hold. For example, the two graphs G_1 and G_2 shown in Figure 7.8 have no perfect matchings; this follows since only one of the vertices v_2 or v_3 or v_4 can be matched with v_1.



(a) G_1 (b) G_2

Figure 7.8: Graphs with no perfect matchings.

Definition. *A component of G is called an* ***odd component*** *if it has odd number of vertices, and it is called an* ***even component*** *if it has an even number of vertices. The number of odd components in G is denoted by $\theta(G)$.*

Lemma 7.6. *If $n(G)$ is even and $S \subseteq V(G)$, then $\theta(G-S)$ and $|S|$ have the same parity.*

Proof. Exercise. $\square$

We can now proceed to state and prove a theorem which characterizes the graphs with perfect matchings.

Theorem 7.7 (Tutte, 1947)**.** *A graph G has a perfect matching iff*

(Tutte) $\theta(G-S) \leq |S|$, *for every* $S \subseteq V(G)$.

Proof. (1) G has a perfect matching $\Rightarrow$ G satisfies (Tutte).

Let M be a perfect matching in G and let $S = \{v_1, v_2, \ldots, v_s\} \subseteq V(G)$. Let $G_1, \ldots, G_t$ be the odd components of $G-S$, where $t = \theta(G-S)$. Since, G_i $(1 \leq i \leq t)$ has odd number of vertices, at least one vertex u_i of G_i is matched under M with a vertex v_i of S; that is $(u_1, v_1), \ldots, (u_t, v_t) \in M$. Since M is a matching, v_i's are all distinct. Hence, $|S| \geq t = \theta(G-S)$.

(2) G satisfies (Tutte) $\Rightarrow$ G has a perfect matching.

We give a proof that is due to Anderson (1971). It is by induction on n. We first observe that n is even: By taking $S = \emptyset$ in (Tutte) we deduce that $\theta(G) = \theta(G-\emptyset) \leq |\emptyset| = 0$. That is, G has no odd components. Hence, n is even. We now proceed to prove the implication.

Basic step $n = 2$: Since G satisfies (Tutte), $G \simeq K_2$ and it has a perfect matching.

Induction step: We make two cases.

Case 1: $\theta(G-S) \leq |S| - 1$, for every proper subset $S \subseteq V(G)$.

By Lemma 7.6, $\theta(G-S) \neq |S|-1$, for any S. So,

$$(P_1) \qquad \theta(G-S) \leq |S|-2, \text{ for every proper subset } S \subsetneq V(G).$$

Let $e=(u,v)$ be an edge in G, and let $H=G-\{u,v\}$.

Claim: H satisfies (Tutte).

On the contrary, suppose that H contains a $T \subseteq V(H)$ such that $\theta(H-T) \geq |T|+1$.

Since, $n(H)$ is even, using Lemma 7.6, we conclude that $\theta(H-T) \geq |T|+2$. But then

$$\theta(G-(T \cup \{u,v\})) = \theta(H-T) \geq |T|+2 = |T \cup \{u,v\}|.$$

Since G satisfies (Tutte), $\theta(G-(T\cup\{u,v\})) \leq |T\cup\{u,v\}|$. Hence, $\theta(G-(T\cup\{u,v\})) = |T \cup \{u,v\}|$. This contradicts our assumption. Hence, the claim holds. Therefore, by induction hypothesis, H has a perfect matching, say M. But then $M \cup \{e\}$ is a perfect matching of G.

Case 2: $\theta(G-S) = |S|$, for some proper subset $S \subsetneq V(G)$.

Let $S^* = \{v_1, v_2, \ldots, v_s\}$ be a subset with ***maximum number*** of vertices such that $\theta(G-S^*) = |S^*|$. We look at $G-S^*$. Let $G_1, \ldots, G_s$ be the odd components of $G-S^*$.

Observation 1: $G-S^*$ has no even components.

On the contrary, if D is an even component of $G-S^*$ and $x \in V(D)$, then $D-x$ has at least one odd component and so $\theta(G-(S^* \cup \{x\})) \geq s+1$. Hence, $\theta(G-(S^* \cup \{x\})) = |S^* \cup \{x\}|$, since G satisfies (Tutte). This is a contradiction to the maximality of $|S^*|$.

Observation 2: Any k $(1 \leq k \leq s)$ odd components from $G_1, \ldots, G_s$ are together joined to at least k vertices of S^*; else, G violates (Tutte).

Therefore, by Hall's Theorem 7.3, there exists a matching, say $M_0 = \{(u_1, v_1), \ldots, (u_s, v_s)\}$, where $u_i \in G_i$ $(1 \leq i \leq s)$.

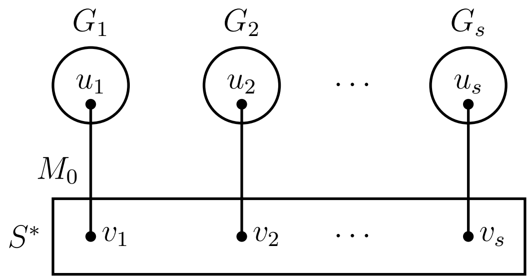


Figure 7.9: The graph G with matching $M_0 = \{(u_1, v_1), \ldots, (u_s, v_s)\}$.

Figure 7.9 shows the structure of G that we have derived until now.

Observation 3: $H_1 = G_1 - u_1$ satisfies (Tutte).

On the contrary, suppose that there exists $T \subseteq V(H_1)$ such that $\theta(H_1 - T) \geq |T| + 1$. Since, $n(H_1)$ is even, using Lemma 7.6, we conclude that $\theta(H_1 - T) \geq |T| + 2$. Let $L_1, L_2, \ldots, L_p$ be the odd component of $H_1 - T$, where $p \geq |T| + 2$. But then $L_1, L_2, \ldots, L_p$ and $G_2, \ldots, G_s$ are odd components of $G - (S^* \cup \{u_1\} \cup T)$. Hence,

$$\begin{aligned} \theta(G - (S^* \cup \{u_1\} \cup T)) &= p + (s - 1), \\ &\geq |T| + 2 + s - 1, \\ &= |T| + 1 + s, \\ &= |(S^* \cup \{u_1\} \cup T)|. \end{aligned}$$

Since G satisfies (Tutte), we conclude that $\theta(G - (S^* \cup \{u_1\} \cup T)) = |S^* \cup \{u_1\} \cup T|$, which is a contradiction to the maximality of $|S^*|$. Hence the observation is proved.

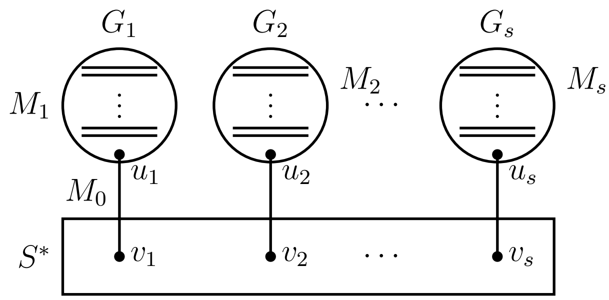


Figure 7.10: G with perfect matchings $M_0 \cup M_1 \cup \ldots \cup M_s$.

Therefore, by induction hypothesis, H_1 has a perfect matching, say M_1. Similarly, $G_2 - \{u_2\}, \ldots, G_s - \{u_s\}$ contain perfect matchings, say $M_2, \ldots, M_s$ respectively. But then $M_1 \cup M_2 \cup \cdots \cup M_s$ is a perfect matching of G; see Figure 7.10. □

Although Tutte's theorem characterizes the graphs with a perfect matching, it is hard to verify Tutte's condition and conclude that a given graph G has a perfect matching, because we have to verify (Tutte) for 2^n subsets of $V(G)$. Hence, there have been several results proved by various mathematicians which say that a given graph G has a perfect matching if G satisfies a certain property (P)(where (P) is easily verifiable). In fact, the first result on perfect matchings was obtained by Petersen (1891) which preceded Tutte's theorem. However, we can easily deduce Petersen's result using Tutte's theorem.

Corollary. *If G is a $(k-1)$-edge-connected k-regular graph with n even, then G has a perfect matching.*

Proof. It is enough if we verify that G satisfies (Tutte). Let $S \subseteq V(G)$ and $G_1, G_2, \ldots, G_t$ be the odd components of $G - S$. Since, G is $(k-1)$-edge-connected, $|[S, V(G_i)]| \geq k - 1$, for every $i, 1 \leq i \leq t$.

Claim: $|[S, V(G_i)]| \geq k$, for every $i, 1 \leq i \leq t$.

On the contrary, suppose that for some j, $1 \leq j \leq t$, $|[S, V(G_j)]| = k - 1$. Then

$$\begin{aligned} \sum_{v \in V(G_j)} deg_{G_j}(v) &= 2m(G_j) + k - 1 \\ \Rightarrow \quad k|V(G_j)| &= 2m(G_j) + k - 1, \text{ since } G \text{ is } k\text{-regular.} \\ \Rightarrow \quad k(|V(G_j)| - 1) &= 2m(G_j) - 1. \end{aligned}$$

While right hand side is an odd integer, left hand side is an even integer since $|V(G_j)|$ is odd. This contradiction proves the claim.

We next estimate the number of edges in $[S, V - S]$ in two different ways.

$$\begin{aligned} (1) \qquad |[S, V - S]| &\leq k|S|, \text{ since } G \text{ is } k\text{-regular.} \\ (2) \qquad |[S, V - S]| &= \sum_{i=1}^{t} |[S, V(G_i)]|, \text{ since } V - S = V(G_1) \cup \cdots \cup V(G_t) \\ &\geq \sum_{i=1}^{t} k, \text{ by the above claim} \\ &= tk. \end{aligned}$$

(1) and (2) imply that $t \leq |S|$. Hence, G satisfies (Tutte) and it has a perfect matching. $\square$

Corollary (Petersen, 1891)**.** *If G is a 2-edge-connected 3-regular graph, then G has a perfect matching.* $\square$

However, every connected 3-regular graph does not have a perfect matching; see Figure 7.8. Hence, we cannot strengthen Petersen's result by assuming that G is connected and 3-regular.

7.5 Greedy and approximation algorithms (Optional)

Since finding $\beta_0(G)$ and $\alpha_0(G)$ are known to be hard problems, there have been attempts to describe algorithms which take G as an input and then output a vertex-cover (or an independent set) with $c\beta_0(G)$ (or $c\alpha_0(G)$) number of vertices where c is a constant. Obviously, in the case of $\beta_0(G)$, $c \geq 1$, and in the case of $\alpha_0(G)$, $c \leq 1$. There is much research to describe algorithms to output vertex covers containing $c\beta_0(G)$ vertices with c as small as possible (and in the case of independent sets with c as large as possible). The constant c is called the ***approximation-factor*** and it is used to measure the efficiency of an algorithm. Obviously among the two algorithms A_1 and A_2 that output $c_1\beta_0(G)$ and $c_2\beta_0(G)$ vertices respectively, where $c_1 < c_2, A_1$ is a better algorithm.

What we have learned in the previous sections is enough for us to describe two such algorithms.

An approximate vertex-cover algorithm with approximation factor 2.

Input: A graph G on n vertices, $M \leftarrow \emptyset$, $K \leftarrow \emptyset$.

Output: A maximal matching M of G and a vertex-cover K of G.

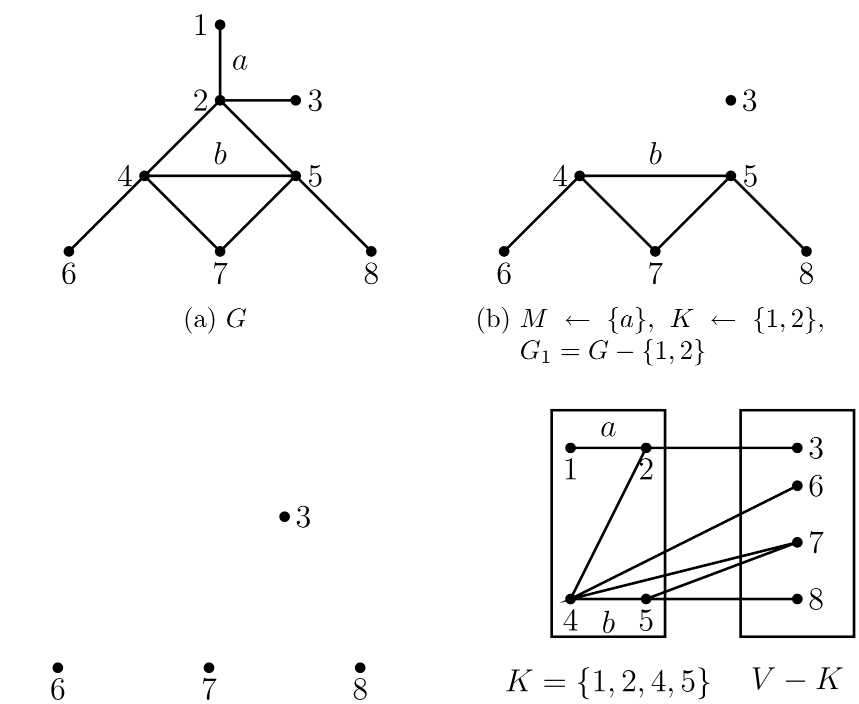
- While $E(G) \neq \emptyset$ do
  - Select any edge $e(u, v)$ from $E(G)$.
  - $M \leftarrow M \cup \{e\}$, $K \leftarrow K \cup \{u, v\}$.
  - $G \leftarrow G - \{u, v\}$.
- end while.
- Output M and K.

We show that $|K| \leq 2\beta_0(G)$.

Theorem 7.8 (Correctness of the algorithm)**.** *Any output K is a vertex-cover of G with $|K| \le 2\beta_0(G)$.*

Proof. At the outset observe that M is indeed a maximal matching. Since at every step after including $e(u, v)$ in M we have deleted all the edges incident to u or v. Since M is a maximal matching, the set of all M-unsaturated vertices $V - K$ is an independent set. That is, every edge in G is incident with a vertex in K. Hence, K is a vertex-cover such that $|K| \le 2|M| \le 2\alpha_1(G) \le 2\beta_0(G)$; since $\alpha_1(G) \le \beta_0(G)$, for any graph G (see the observation made in the beginning of the Section 7.4). $\square$

An illustration for the algorithm:



(a) G

(b) $M \leftarrow \{a\}$, $K \leftarrow \{1, 2\}$, $G_1 = G - \{1, 2\}$

(c) $M \leftarrow \{a, b\}$, $K \leftarrow \{1, 2, 4, 5\}$, $G_2 = G - \{1, 2\} - \{4, 5\}$

Figure 7.11: A graph G with a maximal matching $\{a, b\}$ generated by the greedy algorithm.

- Notice that if we had selected the edge (4,6) instead of (4,5) in the second step, we would have generated a matching with 3 edges which is also a maximal matching.

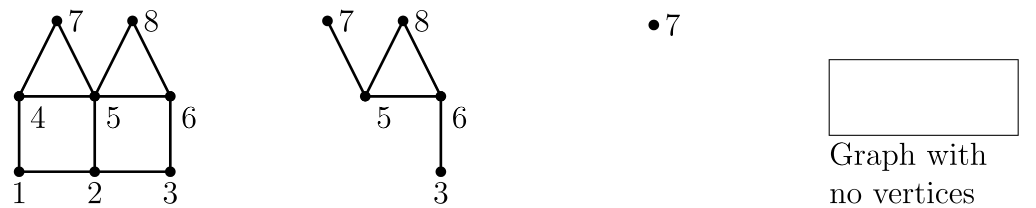
A greedy algorithm to output a maximal independent set of vertices.

Input: A graph G, $I \leftarrow \emptyset$.

Output: A maximal independent set I of vertices.

- While $V(G) \neq \emptyset$ do
  - Select any vertex (arbitrarily) say v from $V(G)$.
  - $I \leftarrow I \cup \{v\}$, $G \leftarrow G - \{v\} - N_G(v)$
- end while.
- Output I.

An illustration:



(a) G; $I \leftarrow \emptyset$ (b) $I \leftarrow \{1\}$, $G_1 = G - \{1\} - \{2,4\}$. (c) $I \leftarrow \{1,6\}$, $G_2 = G_1 - \{6,3,5,8\}$. (d) $I \leftarrow \{1,6,7\}$, $G_3 = G_2 - \{7\}$.

Figure 7.12: A graph G with maximal independent set $I = \{1,6,7\}$ generated by the greedy algorithm.

Remarks.

- The algorithm indeed outputs an independent set since after selecting a vertex v we delete v and all the vertices adjacent to v.
- The algorithm need not generate a maximum independent set. If we had selected the vertex 3 in G_1 instead of 6 and thereafter had selected 7, we could have generated an independent set $\{1,3,7,8\}$ which is a maximum independent set.

Theorem 7.9. *The algorithm outputs an independent set I with at least $\dfrac{n}{1+\Delta(G)}$ vertices.*

Proof. The algorithm indeed outputs a maximal independent set of vertices, since at every step after selecting vertex v, we delete v and all its neighbors, and in the next step, we select a new vertex from $G - v - N(v)$. Suppose $I = \{v_1, v_2, \ldots, v_t\}$ is the output. After selecting v_t, we are left with no more vertices in the residual graph. So

$$\begin{aligned} n &\leq \underbrace{(1+\Delta)+\cdots+(1+\Delta)}_{\text{sum } t \text{ times}}, \\ &\qquad \text{since in each step we have deleted at most } 1+\Delta \text{ vertices.} \\ &= t(1+\Delta) \\ \text{Hence,} \qquad |I| &= t \geq \frac{n}{1+\Delta}. \quad \square \end{aligned}$$

Exercises

1. Let G be a simple bipartite graph. Show:
   (a) $|E(G)| \leq \alpha_0(G)\beta_0(G)$.
   (b) If $|E(G)| = \alpha_0(G)\beta_0(G)$, then G is a complete bipartite graph;
2. For a bipartite graph G, show that the following statements are equivalent:
   (a) $\alpha_0(H) \geq |V(H)|/2$, for every subgraph H of G.
   (b) $\alpha_0(H) = \beta_1(H)$, for every subgraph H of G with $\delta(H) > 0$.
3. Let G be a complete n-partite graph. Prove the following:
   (a) $\beta_0(G) = \delta(G) = k_0(G) = k_1(G)$
   (b) G is Hamilton iff $|V(G)| \leq 2\beta_0(G)$.
4. Prove that for any graph G, the following inequalities are equivalent.
   (a) $\beta_0(G) + \beta_0(G^c) \geq |V(G)|$.
   (b) $\alpha_0(G) + \alpha_0(G^c) \leq |V(G)|$.

(c) $\beta_0(G) + \beta_0(G^c) \geq \alpha_0(G) + \alpha_0(G^c)$.

(d) $\beta_0(G) \geq \omega(G)$,
where $\omega(G) = \max\{|V(H)| : H$ is a complete subgraph of $G\}$.

5. For any graph G, show:
   (a) $\beta_0(G) \geq \delta(G)$.
   (b) $\beta_0(G) \geq \omega(G) - 1$.

6. Prove or disprove: In any tree there is a maximal independent set of vertices containing all the vertices of degree one.

7. Find the vertex-covering number of $K_{1,t} + K_{1,t}$ $(t > 1)$.

8. Find the minimum edge covering number β_1 of the d-cube Q_d. Justify your answer. And give an example of a minimum edge covering set of Q_d.

9. Find β_0 and β_1 of $K_{2,4,6,\ldots,2k}$.

10. Find α_0 and β_1 of $C_{2m+1} + C_{2n}$ and $K_n + C_{2m+1}$.

11. Find α_0, β_0, α_1 and β_1 of $K_{1,2,\ldots,n}$.

12. Find the matching number of $C_5 + C_7$.

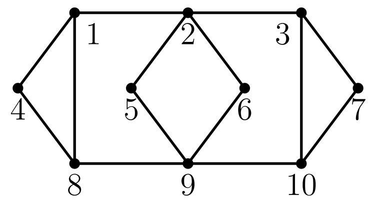
13. Let M, N be disjoint matchings of a graph G with $|M| > |N|$. Show that there are disjoint matchings M' and N' of G such that $|M'| = |M| - 1$ and $|N'| = |N| + 1$ and $M' \cup N' = M \cup N$.

14. Draw a connected 4-regular graph with no perfect matching on n-vertices where n is an even integer. What is the minimum value of n. Justify your graph indeed has no perfect matching.

15. For each $k \geq 2$, find an example of a k-regular simple graph which has no perfect matching.

16. Prove or disprove:
    (a) The graph shown in the figure below contains a perfect matching.
    (b) A tree has at most one perfect matching.

17. A tree T has a perfect matching iff $O(T - v) = 1$, for every vertex v.



18. Show that no tree ($n \geq 3$) with every vertex of degree 1 or 3 has a perfect matching.

19. Draw a 3−regular connected simple graph with a perfect matching that has either a cut-vertex or a cut edge.

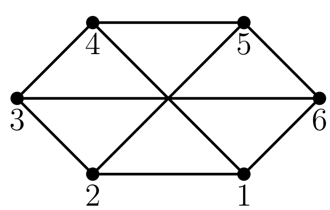
20. Let $n > 1$ be an integer. Prove or disprove:
    (a) If n is even, $K_{n,n,n}$ has a perfect matching.
    (b) If n is odd, $K_{n,n,n}$ has no perfect matching.

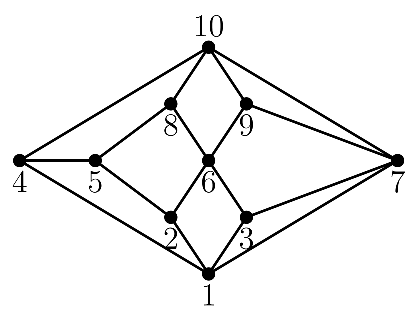
21. Draw a connected simple bipartite graph $G[U, V]$ with minimum degree at least 3 and $|U| = |V| = 7$ which has no matching saturating every vertex in U.

22. Prove or disprove: If $G[X, Y]$ is a simple connected bipartite graph such that the degree of every vertex in X is at least 3 and the degree of every vertex in Y is at most 3, then G has a matching saturating every vertex in X.

23. Illustrate the greedy algorithm to get a maximal independent set for the simple graph whose vertices are a, b, c, d, e, f and whose edges are (a, b), (b, c), (c, d), (d, e), (e, f), (f, a), (b, f), (c, e).

24. Apply greedy algorithm to get a maximal matching and a vertex cover of the graphs shown below.





Module 8 Vertex-colorings

Contents

The concept of vertex coloring of a graph can be used to model many scheduling problems, optimal assignment of channels to radio stations and optimal assignment of spectrum frequencies to mobile operations. However, the topic originated with the following "map coloring conjecture".

At most four colors are required to color any map of a country so that adjacent states receive different color.

This problem was first raised by Francis Guthrie in a letter to De Morgan in 1852. For a long time, it was a fascinating open problem among mathematicians. Arthur Cayley (1821 – 1852) and G.D. Birkhoff (1884 – 1944) made it known to a wider community by their presentations and innovative contributions. It was settled in affirmative by K. Appel, W. Haken and J. Kosh in 1977.

8.1 Basic definitions

Definition. *Any function $f : V(G) \to \{1, 2, \ldots, k\}$ is called a* ***k-vertex-coloring*** *of a graph G if $f(u) \neq f(v)$, for any two adjacent vertices $u, v \in G$. The integers $1, 2, \ldots, k$ are called the* ***colors****. Equivalently, a k-vertex-coloring of G is a partition $(V_1, \ldots, V_k)$ of $V(G)$ where each V_i is an independent set of vertices in G.*

Remarks.

- Every graph G admits a n-vertex-coloring. However, given k, G may not admit a k-vertex-coloring.
  Figure 8.1 shows various colorings of $K_4 - e$.
- If G admits a k-vertex-coloring, then it also admits a $(k+1)$-vertex-coloring.

The above remarks motivate the following important concept.

Definition. *The minimum integer k such that G admits a k-vertex-coloring is called the* ***vertex-chromatic-number*** *of G. It is denoted by $\chi(G)$ or $\chi_0(G)$. In other*

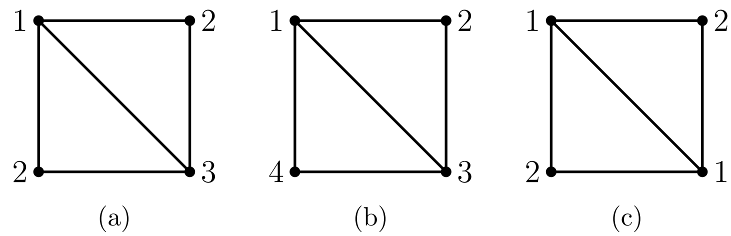


Figure 8.1: (a) shows a 3-vertex-coloring, (b) shows a 4-vertex-coloring and (c) shows a coloring which is not a 2-vertex-coloring.

words, $\chi(G)$ is the minimum number of colors required to color the vertices such that no two adjacent vertices of G receive the same color.

The following table shows the vertex-chromatic number of some standard graphs.

G	K_n	K_n^c	Bipartite graph	Petersen graph
$\chi(G)$	n	1	2	3

Table 8.1: Vertex-chromatic number of some standard graphs.

Remarks.

- If H is the underlying simple graph of G, then $\chi(H) = \chi(G)$.
- If $G_1, G_2, \dots, G_t$ are the components of G, then
$$\chi(G) = \max\{\chi(G_1), \chi(G_2), \dots, \chi(G_t)\}.$$
- If $B_1, B_2, \dots, B_p$ are the blocks of G, then
$$\chi(G) = \max\{\chi(B_1), \chi(B_2), \dots, \chi(B_p)\}.$$
- In view of the above remarks, the study of vertex-colorings may be restricted to simple connected graphs.

8.2 Cliques and chromatic number

Definition. *A subgraph Q of a graph G is called a **clique**, if any two vertices in Q are adjacent. A **maximal clique** is a clique that is not a subgraph of any other clique. The **clique number** of G is the integer $\omega(G) = \max\{|V(Q)| : Q \text{ is a clique in } G\}$.*

If H is a subgraph of G, then obviously $\chi(H) \leq \chi(G)$. Hence,

- For any graph G, $\chi(G) \geq \omega(G)$.

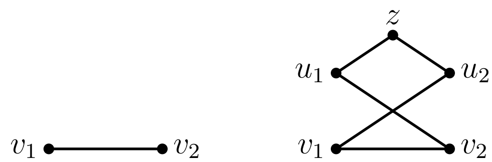
• Mycielski's theorem

There has been intense research to obtain upper bounds for $\chi(G)$ in terms of $\omega(G)$ alone. Questions like for which graphs $\chi(G) = \omega(G)$ or $\chi(G) = \omega(G) + 1$ or $\chi(G) \leq f(\omega(G))$ for a function f, have received wide attention. Mycielski (1955) showed that given any k, there exists a graph G with $\omega(G) = 2$ and $\chi(G) = k$. This result indicates that there is no upper bound for an arbitrary class of all graphs as a function of ω alone.

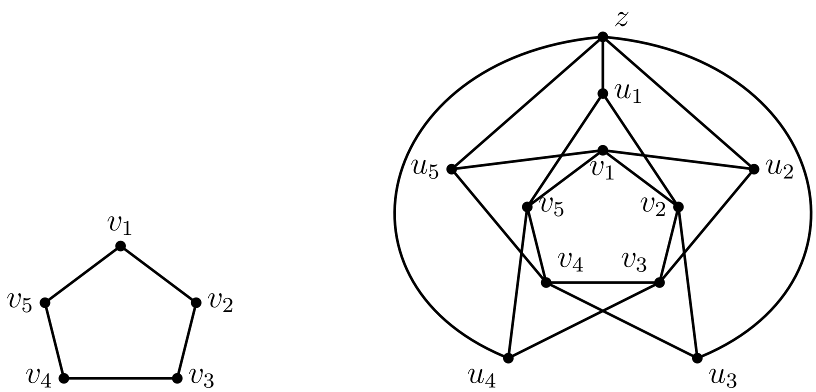
Theorem 8.1. *For any given integer k (≥ 1), there exists a triangle-free graph G_k with vertex-chromatic number k. (Triangle-free $G :=$ G has no K_3 as a subgraph.)*

Proof. G_k is constructed by induction on k. For $k = 1$ and 2, K_1 and K_2 are the required graphs. Next suppose that we have constructed a triangle-free graph G_k on p vertices $v_1, v_2, \ldots, v_p$. Construct G_{k+1} from G_k as follows.

1. Add $p + 1$ new vertices $u_1, u_2, \ldots, u_p$ and z to G_k.
2. Join u_i ($1 \leq i \leq p$) to all those vertices to which v_i is adjacent in G_k.
3. Join z to every u_i.
   Figure 8.2 shows this construction of G_3 from G_2 ($= K_2$) and G_4 from G_3 ($= C_5$).



(a) $G_2 \to G_3$



(b) $G_3 \to G_4$

Figure 8.2: Mycielski's construction.

We shall prove that G_{k+1} is triangle-free and that $\chi(G_{k+1}) = k+1$.

(a) G_{k+1} is triangle-free.

On the contrary, suppose that G_{k+1} contains a triangle $\mathcal{T}$. Since G_k is triangle-free, one of the vertices of $\mathcal{T}$ is z or u_i (for some i). If $z \in \mathcal{T}$, then the other two vertices of $\mathcal{T}$ are u_g and u_h (for some g and h). This is a contradiction, since no two vertices in $\{u_1, \ldots, u_p\}$ are adjacent. If $u_i \in \mathcal{T}$ and $z \notin \mathcal{T}$, then the other two vertices of $\mathcal{T}$ are v_g and v_h (for some g and h). But then, v_i, v_g and v_h are the vertices of a triangle in G_k, which again is a contradiction. Hence, G_{k+1} is triangle-free.

(b) $\chi(G_{k+1}) = k+1$.

Since, G_k is a subgraph of G_{k+1}, we have $\chi(G_{k+1}) \geq \chi(G_k) = k$. If $f : V(G_k) \to \{1, 2, \ldots, k\}$ is a k-coloring of G_k, then $g : V(G_{k+1}) \to \{1, 2, \ldots, k+1\}$ defined by

$$\begin{aligned} g(v_i) &= f(v_i), 1 \leq i \leq p, \\ g(u_i) &= f(v_i), 1 \leq i \leq p, \\ g(z) &= k+1, \end{aligned}$$

is a $(k+1)$-coloring of G_{k+1}. Hence, $\chi(G_{k+1}) \leq k+1$.

Next if possible suppose that $\chi(G_{k+1}) = k$, and let $h : V(G_{k+1}) \to \{1, 2, \ldots, k\}$ be a k-coloring of G_{k+1}. Without loss of generality, let $h(z) = k$. So, $h(u_i) \neq k$ for $i = 1, 2, \ldots, k$. Define $g : \{v_1, v_2, \ldots, v_p\} \to \{1, 2, \ldots, k-1\}$ by

$$g(v_i) = \begin{cases} h(v_i), & \text{if } h(v_i) \neq k \\ h(u_i), & \text{if } h(v_i) = k \end{cases}$$

It is easy to verify that g is a $(k-1)$-vertex-coloring of G$_k$. This is a contradiction to $\chi(G_k) = k$.

Hence, we conclude that $\chi(G_{k+1}) = k+1$. □

8.3 Greedy coloring algorithm

Finding $\chi(G)$ of an arbitrary graph G is a hard problem. So there have been attempts to use greedy and approximate algorithms and obtain tight bounds. However, it has been shown that there can be no approximate algorithm for finding $\chi(G)$ within a factor of $n^{1-\epsilon}$, for any $\epsilon > 0$, where n is the number of vertices in G. This deep result shows that finding the vertex-chromatic number of an arbitrary class of graphs is impossibly hard. Despite this negative result, greedy algorithm does use nearly χ colors if the input graph belongs to several interesting classes of graphs. This sections explores these aspects.

Greedy algorithm (Sequential algorithm/An example of an on-line algorithm.)

Input: A graph G and an ordering $(v_1, v_2, \ldots, v_n)$ of its vertices.

Output: A k-vertex-coloring of G, for some k.

Step 1: Color v_1 with 1.

Step 2: Having colored $v_1, v_2, \ldots, v_i$ with t colors, say $1, \ldots, t$, color v_{i+1} with the following policy. Let T be the set of colors used to color the vertices adjacent with v_i, then

$$color(v_{i+1}) = \begin{cases} \min(\{1, 2, \ldots, t\} - T), & \text{if } T \neq \{1, 2, \ldots, t\} \\ t + 1, & \text{if } T = \{1, 2, \ldots, t\}. \end{cases}$$

Step 3: Stop when v_n is colored.

An illustration:

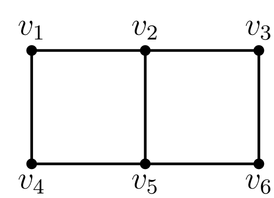
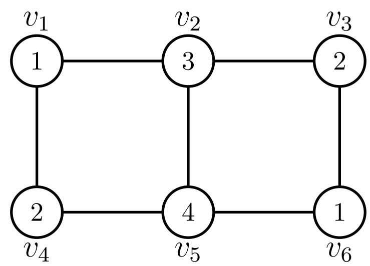
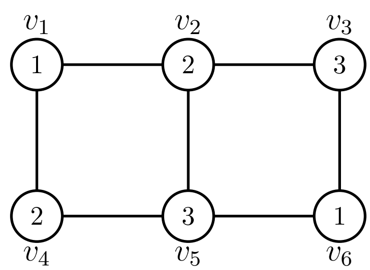


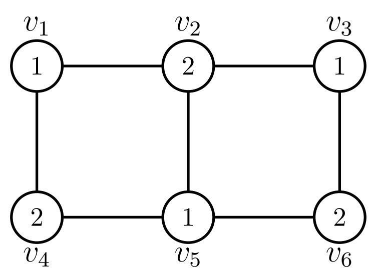
Figure 8.3: A graph G.



(a) An output of the greedy algorithm with input $(G; (v_1, v_6, v_3, v_4, v_2, v_5))$. The number inside the circle indicates its colors.



(b) An output of the greedy algorithm with input $(G; (v_1, v_2, v_6, v_3, v_5, v_4))$.



(c) An output of the greedy algorithm with input $(G; (v_1, v_3, v_5, v_2, v_4, v_6))$. Since $\chi(G) = 2$, with this choice of vertex ordering the algorithm uses the minimum number of colors.

Figure 8.4: Outputs of the greedy algorithm when three different vertex orderings are given as input.

Remarks.

- Algorithm is called a greedy algorithm since we use the first available color at every step.
- It is called a sequential algorithm since the input is a sequence of the vertices and the vertices are colored sequentially.

- It is an example of an on-line algorithm since at every step we use the available data to color the vertex and do not change this color subsequently.

- The illustration indicates that the number of colors used by the algorithm depends on the ordering of the vertices.

Theorem 8.2. *The greedy algorithm uses at most* $\Delta(G) + 1$ *colors.*

Proof. While coloring v_{i+1}, under Step 3, there are at most $\Delta(G)$ neighbours of v_{i+1} in $\{v_1, v_2, \ldots, v_i\}$, that is $|T| \leq \Delta(G)$. Hence, $color(v_{i+1}) \leq 1 + \Delta(G)$. This holds for every i. Hence, the result. □

Corollary. *For any graph* G, $\chi(G) \leq 1 + \Delta(G)$. □

Remark. If the vertices of G can be ordered $(v_1, v_2, \ldots, v_n)$ so that every v_i has at most $k-1$ neighbours in $\{v_1, v_2, \ldots, v_{i-1}\}$, then the greedy algorithm uses at most k colors.

• Coloring of chordal graphs (Optional)

Theorem 8.3. *We can color any chordal graph* G *with* $\omega(G)$ *colors using the greedy algorithm.*

Proof. Let S be a PEO of the vertices of G; see Theorem 4.7. Under Step 1 of the greedy algorithm, choose the reverse ordering of S, say $(v_1, v_2, \ldots, v_n)$. By the PEO property, v_{i+1} and its neighbors in $\{v_1, v_2, \ldots, v_i\}$ form a complete subgraph in G. Hence at the $(i+1)$-th iteration, there are at most $\omega(G) - 1$ colors appearing on the neighbors of v_{i+1}. Hence the greedy algorithm uses at most $\omega(G)$ colors. □

An illustration:

Corollary. *For any chordal graph* G, $\chi(G) = \omega(G)$. □

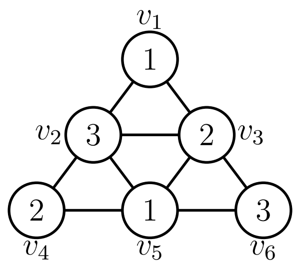


Figure 8.5: A graph G with PEO $(v_1, v_4, v_6, v_2, v_3, v_5)$ and its reverse order $(v_5, v_3, v_2, v_6, v_4, v_1)$ as the input.

We observed in Theorem 8.2 that $\chi(G) \le 1 + \Delta(G)$, for any graph G. Clearly, $\chi(C_{2n+1}) = 3 = 1 + \Delta(C_{2n+1})$ and $\chi(K_n) = n = 1 + \Delta(K_n)$. Surprisingly, Brooks (1941) showed that these are the only classes of graphs for which the upper bound is attained.

• Brooks theorem (Optional)

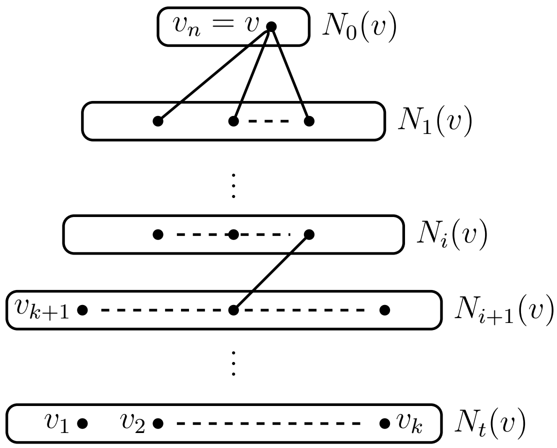
Theorem 8.4 (Brooks, 1941)**.** *If $G \notin \{C_{2p+1}, K_n\}$, then $\chi(G) \le \Delta(G)$.*

Proof. (Lovasz, 1975) Without loss of generality, we assume that G is 2-connected. As remarked earlier, the idea of the proof is to get an ordering $(v_1, v_2, \ldots, v_n)$ of the vertices of G such that every v_i has at most $\Delta - 1$ neighbors in $(v_1, v_2, \ldots, v_{i-1})$.

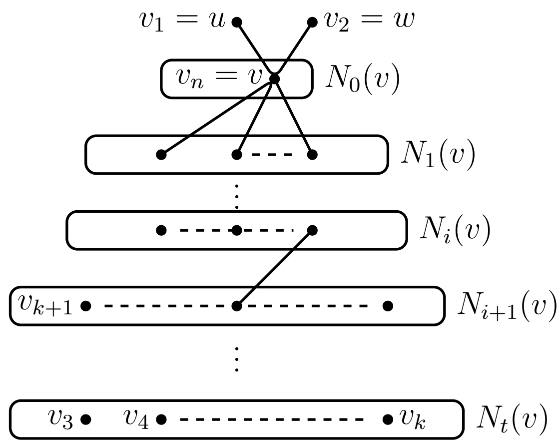
Case 1: G is not regular, that is $\Delta(G) \ne \delta(G)$.

Let v be a vertex of minimum degree and let $t = \max\{d(v,x) : x \in V(G)\}$. Let $N_i(v) = \{x \in V(G) : d(v,x) = i\}$, for $i = 0, 1, \ldots, t$. Clearly, $V(G) = \cup_{i=0}^{t} N_i(v)$, and every $x \in N_i(v)$ is adjacent to some vertex in $N_{i-1}(v)$, for every i, $i = 1, \ldots, t$. Let $S = (N_t^+(v), N_{t-1}^+(v), \ldots, N_0(v))$ be an ordering of the vertices of G where $N_i^+(v)$ denotes a sequence of the vertices of $N_i(v)$ ordered arbitrarily; see Figure 8.6a.

We apply greedy algorithm with S as the input. Every vertex $v_p (\ne v)$ in $N_{i+1}(v)$ is adjacent with a vertex in $N_i(v)$, $0 \le i \le t-1$. Hence when v_p is getting



(a) Vertex ordering for Case 1



(b) Vertex ordering for Case 2

Figure 8.6: An ordering of the vertices of G.

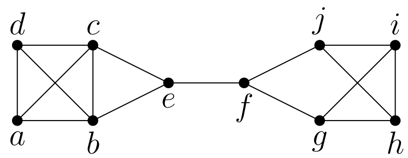
colored there are at most at most $\Delta - 1$ neighbors in $\{v_1, v_2, \ldots, v_{p-1}\}$. Therefore, v_p is colored with one of the colors in $\{1, 2, \ldots, \Delta\}$. Since $v_n (= v)$ has degree $\delta(G) < \Delta(G)$, v_n too has at most $\Delta - 1$ neighbors in $\{v_1, v_2, \ldots, v_{n-1}\}$. Therefore v_n is colored with one of the colors in $\{1, 2, \ldots, \Delta\}$.

We illustrate the above case with an example.

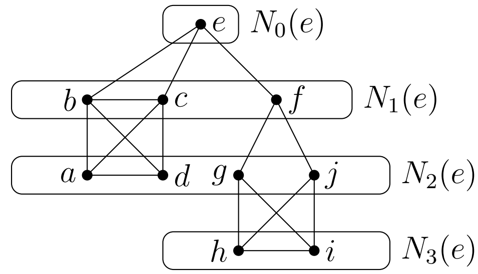
Illustration: Consider the graph shown in Figure 8.7. The graph G is a non-regular graph with $\Delta(G) = 4$. By proceeding as in Case 1 with e as the initial vertex, the neighborhood decomposition of the graph yields an ordering of the vertices (chosen arbitrarily within each level) given by $(h, i, a, d, g, j, b, c, f, e)$. Application of the greedy algorithm to this vertex ordering, gives a 4-coloring $(1, 2, 1, 2, 3, 3, 3, 4, 1, 2)$ to the graph as desired.

Case 2: G is d-regular for some d; $2 \leq d \leq n - 2$, since $G \neq K_n$.

If $d = 2$, then G is an even cycle, and so $\chi(G) = 2$. Next assume that $3 \leq d \leq n - 2$. Note that the greedy algorithm uses at most Δ colors in this case



(a) Input Graph G



(b) An ordering of G

Figure 8.7: A graph G with $\Delta = 4$ with an ordering $(h, i, a, d, g, j, b, c, f, e)$ of the vertices.

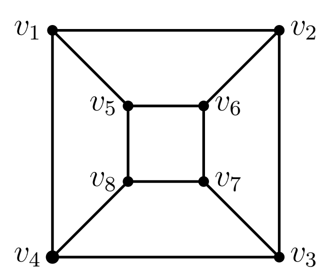
too, if we can get an order $(v_1, v_2, \ldots, v_n)$ of the vertices such that every v_i $(\neq v_n)$ is adjacent with some vertex succeeding v_i and that v_n has at least two neighbors of the **same** color in $(v_1, v_2, \ldots, v_{n-1})$. Such an ordering is achieved by appealing to the following claim (whose proof is left as an exercise).

Claim: In any 2-connected k-regular graph $(3 \leq k \leq n-2)$, there exist three vertices u, v and w such that $(u, v), (v, w) \in E(G)$, $(u, w) \notin E(G)$ and $G - \{u, w\}$ is connected.

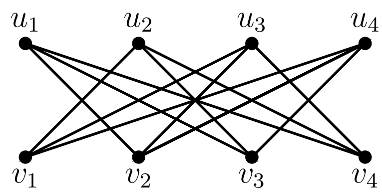
Let u, v, w be as in the claim and let $S = (v_1 = u, v_2 = w, N_t^+(v), \ldots, N_0^+(v))$, where $N_i(v) = \{x \in V(G - \{u, w\}) : d(x, v) = i\}$, $i = 0, 1, 2, \ldots, t$ and $N_i^+(v)$ is the set of vertices in $N_i(v)$ ordered arbitrarily; see Figure 8.6b. With S as the input, the greedy algorithm colors both the vertices v_1 and v_2 with color 1. As in case 1, every vertex v_i $(3 \leq i \leq n-1)$ is adjacent with some vertex succeeding v_i in S. Hence when v_i is getting colored it has at most $\Delta - 1$ colors appearing on its neighbors in $(v_1, v_2, \ldots, v_{i-1})$. Therefore, the greedy algorithm colors v_i with one of the colors in $\{1, 2, \ldots, \Delta\}$. Finally when v_n is getting colored it has at most $\Delta - 1$ colors appearing on its neighbors in $(v_1, v_2, \ldots, v_{n-1})$ since v_1 and v_2 are its neighbors and both are colored 1. Therefore, the greedy algorithm colors v_n too with one of the colors in $\{1, 2, \ldots, \Delta\}$. $\square$

Exercises

1. Find the vertex chromatic number of :
   (a) a tree.
   (b) a graph with exactly one cycle.
   (c) the complement of a cycle.
   (d) $C_{2s} + C_{2t+1}$
   (e) Petersen graph.
   (f) $K_{n_1,n_2,\ldots,n_s}$.
2. Prove: $\chi(G_1 + G_2) = \chi(G_1) + \chi(G_2)$.
3. For every graph G, show that $\frac{n(G)}{\alpha_0(G)} \leq \chi(G) \leq n + 1 - \alpha_0(G)$.
4. Show that every graph with $\chi(G) = k$, contains
   (a) at least $k(k-1)/2$ edges, and
   (b) a cycle of length at least k.
5. Find $n(G_k)$, where G_k is the k-vertex-chromatic, triangle-free graph constructed using Mycielski's method.
6. Draw a graph having maximum number of edges with $\chi(G) = 4$, $n(G) = 10$.
7. Let G be the graph shown below: Order the vertices of G in 3 different ways, say S_1, S_2 and S_3 so that the greedy algorithm
   (a) uses 2 colors, if S_1 is the input,
   (b) uses 3 colors, if S_2 is the input, and
   (c) uses 4 colors, if S_3 is the input.



8. Order the vertices of the graph shown below, in such a way that the greedy algorithm uses 4 colors.



9. Let G be a simple graph such that $\delta(H) \leq k$, for every subgraph H of G (such a graph is called a ***k-degenerate graph***). Order the vertices of G so that if one applies greedy algorithm to color the vertices of G, then it uses at most $k+1$ colors.

10. There are 7 chemicals A, B, C, D, E, F , G. For safety reasons, the following pairs of chemicals cannot be stored in one room: $A\&B$, $A\&C$, $A\&G$, $B\&C$, $B\&D$, $C\&D$, $D\&E$, $D\&F$, $E\&F$, $E\&G$, $F\&G$. If the problem is to find the minimum number of rooms (say k) required to store all the chemicals, how would you model it as a graph theoretical problem and find k.

A graph G is said to be ***k-critical*** if

- $\chi(G) = k$, and
- $\chi(H) < k$, for every proper subgraph H of G.

---

11. If G is k-critical, then show that G is connected and that $\delta(G) \geq k-1$.

12. If S is a set of independent vertices in a k-critical graph, then what is the chromatic number of $G - S$.

13. If G_1 and G_2 are simple graphs, show that $G_1 + G_2$ is a critical graph iff G_1 and G_2 are critical.

A partition $(V_1, V_2, \ldots, V_p)$ of $V(G)$ is called a ***clique partition*** of G if every induced subgraph $[V_i]$, $1 \leq i \leq p$, is a clique in G. The minimum integer k such that G admits a clique partition is called the ***clique partition number*** of G; it is denoted by $\theta_0(G)$.

---

14. Show that:

    (a) $\theta_0(G) = \chi(G^c)$.

    (b) $\theta_0(G_1 + G_2) = \max\{\theta_0(G_1), \theta_0(G_2)\}$.

    (c) $\theta_0(G) \geq \alpha_0(G)$.

Module 9 Edge colorings

Contents

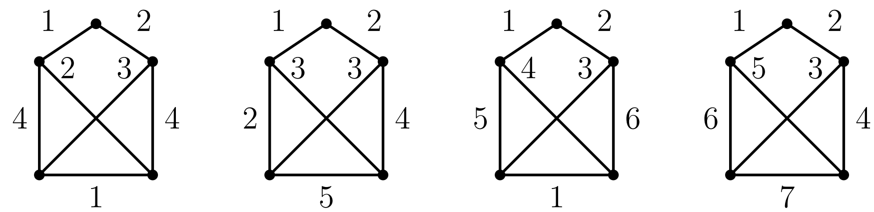
9.1 Introduction and Basics

An obvious variation of vertex-coloring is edge-coloring where the edges are colored rather than vertices. The concept is useful to model many scheduling problems. It also arises in many circuit board problems where the wires connecting a device have to be of different color.

Throughout this chapter, graphs do not have loops (reason will be clear soon) but may have multiple edges.

Definition. *Any function $C : E(G) \rightarrow \{1, 2, \ldots, k\}$ is called a k-**edge-coloring** of G, if $C(x) \neq C(y)$ for any two adjacent edges x and y.*

Equivalently, a k-edge-coloring of G is a partition $(M_1, M_2, \ldots, M_k)$ of $E(G)$ such that each M_i is a matching in G. Throughout, we assume that the edges in M_i are colored i, $1 \leq i \leq k$. See Figures 9.1 and 9.2 for examples of edge-colorings.



(a) $k = 4$ (b) $k = 5$ (c) $k = 6$ (d) $k = 7$

Figure 9.1: A k-edge-coloring of a graph G for $k = 4, 5, 6$ or 7. Notice that there is no k-edge-coloring of G if $k = 1, 2$ or 3.

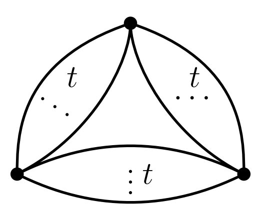


Figure 9.2: An interesting graph for which only edge-coloring is the m-edge-coloring $(m = 3t)$.

Definitions. *Let $C : E(G) \to \{1, 2, \ldots, k\}$ be an edge-coloring of G.*

- *An edge e colored β (that is, $C(e) = \beta$) is called a β**-edge.***
- *Let v be a vertex in G. If there is no β-edge incident with v, then we say that the color β is **missing** (or is **absent**) at v.*
- *If there is a β-edge incident with v, then we say that the color β **appears** (or is **present**) at v.*

Remarks.

- Every graph G admits a m-edge-coloring; color all the edges differently.
- Given k, a graph may not admit a k-edge-coloring.

The above remarks motivate the following concept.

Definition. *The minimum integer k such that G admits a k-edge-coloring is called the **edge-chromatic-number** of G. It is denoted by $\chi_1(G)$.*

- *Thus, $\chi_1(G)$ is the minimum number of colors required to color the edges of G such that no two adjacent edges receive the same color.*
- *The edge-chromatic-number is also called the **chromatic index**.*

Remarks.

- $\chi_1(G) = \chi(L(G))$, if $E(G) \neq \phi$.
- $\chi_1(G) \geq \Delta(G)$, since in any edge-coloring, the colors appearing at any vertex are all distinct. More generally, if C is an edge-coloring and $C(v)$ denotes the number of colors appearing at v, then $C(v) = deg(v)$.
- A natural question is to ask how much $\chi_1(G)$ is bigger than $\Delta(G)$. A remarkable theorem, independently proved by V.G. Vizing (1964) and R. P. Gupta (1966) states that $\chi_1(G) \leq \Delta(G) + 1$, for any simple graph G. We shall prove this theorem soon.

The following table shows the edge-chromatic-number of some elementary graphs. Proofs for the last two entries require some efforts.

G	P_n	C_n, n even	C_n, n odd	Petersen graph	$K_{r,s}$	K_n, n even	K_n, n odd
$\chi_1(G)$	2	2	3	4	$\max\{r,s\}$	$n-1$	n

Table 9.1: A table of edge-chromatic numbers, where $n \geq 3$.

9.2 Gupta-Vizing theorem

Theorem 9.1 (V. G. Vizing (1964), R. P. Gupta (1966))**.** *For any simple graph G,* $\Delta(G) \leq \chi_1(G) \leq \Delta(G)+1$.

Proof. (Indirect Induction/Contradiction method) As observed before, the lower bound is obvious. So, we have to only prove the upper bound. Assume the contrary and let G be a graph having the minimum number of edges among all the graphs with $\chi_1 \geq \Delta+2$. So, $\chi_1(G) \geq \Delta(G)+2$ but $\chi_1(G-e) \leq \Delta(G-e)+1 \leq \Delta(G)+1$, for every edge e.

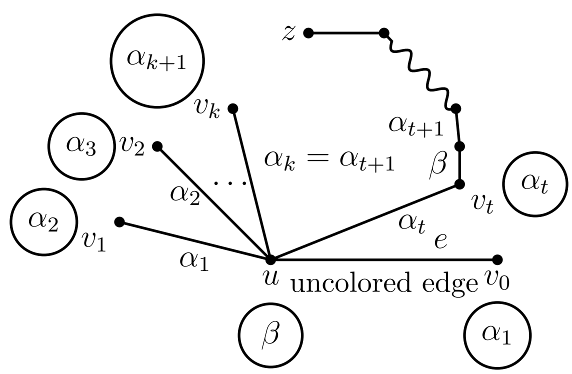


Figure 9.3: Missing colors are shown inside the circles.

Let C be a $(\Delta+1)$-edge-coloring of $G-e$, where e is (u, v_0). Before proceeding further, we remark that:

- At any vertex x at least one color is absent, since there are at most Δ edges incident with x and we have $\Delta + 1$ colors.
- If there exists a $(\Delta + 1)$-edge-coloring C' of $G - e$ such that some color β is absent at u and at v_0, then e can be colored with β to get a $(\Delta+1)$-edge-coloring of G, which provides a contradiction to our choice of G.

Using a recoloring technique, we show that it is always possible to get an edge-coloring C' as described above.

Let α_1 be a color missing at v_0. There is an α_1-edge incident with u; else, by coloring (u, v_0) with α_1 we get a $(\Delta + 1)$-edge-coloring of G. Let (u_1, v_1) be the edge colored α_1; see Figure 9.3.

Let α_2 be a color missing at v_1. There is an α_2-edge incident with u; else, by recoloring (u, v_1) with α_2 and (u, v_0) with α_1 we get a $(\Delta+1)$-edge-coloring of G. So, there must exist an edge say (u, v_2) colored α_2.

Let α_3 be a color missing at v_2. There is an α_3-edge incident with u; else, by recoloring the edges (u, v_2), (u, v_1) and (u, v_0) with colors α_3, α_2 and α_1, respectively we get a $(\Delta + 1)$-edge-coloring of G.

We continue this process to generate:

- a sequence of distinct vertices $v_0, v_1, v_2, \ldots$ incident with u, and
- a sequence of colors $\alpha_1, \alpha_2, \alpha_3, \ldots$ such that the edges $(u, v_1), (u, v_2), \ldots$ are colored $\alpha_1, \alpha_2, \ldots$ respectively, and α_{i+1} is the color missing at $v_i, i \geq 0$.

Since the degree of u is finite, the sequence $v_0, v_1, v_2, \ldots$ is finite. Therefore, there exists some v_t such that its missing color $\alpha_{t+1} \in \{\alpha_1, \ldots, \alpha_t\}$. Let t be the smallest integer such that $\alpha_{t+1} = \alpha_k$, where $\alpha_k \in \{\alpha_1, \ldots, \alpha_t\}$. This is equivalent to saying that there exists an integer k such that (u, v_k) is colored $\alpha_{t+1}(= \alpha_k)$, where $1 \leq k < t$.

Let β be a color missing at u but appearing at v_0. Such a color exists; else we can color (u, v_0) with β to get a $(\Delta + 1)$-edge-coloring of G.

Claim: β appears at every v_i, $0 \leq i \leq t$.

To prove the claim, assume the contrary and let p be the smallest integer such that β is absent at v_p $(1 \leq p \leq t)$. Recolor (u, v_p) with β and edges (u, v_{p-1}), $(u, v_{p-2}), \ldots, (u, v_0)$ with colors $\alpha_p, \alpha_{p-1}, \ldots, \alpha_1$ to get a $(\Delta + 1)$-edge-coloring of G. So the claim holds.

We define a path P as follows:

P is a maximal path with origin v_t and its edges colored alternately β and α_{t+1}. Suppose P terminates at a vertex z. At the outset, observe that the first edge and the last edge of P are colored β; see the claim above. Also $v_t \neq v_k$.

We consider three different cases depending on the location of z, and in each case describe a technique to recolor the edges $G - e$ which leads to a $(\Delta + 1)$-edge-coloring of G.

Case 1: $z \in V(G) - \{v_0, v_1, \ldots, v_t, u\}$.

Interchange the colors β and α_{t+1} of P. After this interchange, there is no β-edge incident with v_t. Recolor (u, v_t) with β, and edges $(u, v_{t-1}), \ldots, (u, v_0)$ with colors $\alpha_t, \ldots, \alpha_1$, respectively. This yields a $(\Delta + 1)$-edge-coloring of G.

Case 2: $z = v_j$, for some j, $0 \leq j \leq t - 1$.

Interchange the colors β and α_{t+1} of P. After this interchange, there is no β-edge incident with v_j. Recolor (u, v_j) with β, and recolor the edges $(u, v_{j-1}), \ldots, (u, v_0)$ with colors $\alpha_j, \ldots, \alpha_1$, respectively. Again we obtain a $(\Delta + 1)$-edge-coloring of G.

Case 3: $z = u$. In this case, the last edge of P is (v_k, u). Interchange the colors β and α_{t+1} of P. After this interchange, (v_k, u) is colored β. Recolor the edges $(u, v_{k-1}), \ldots, (u, v_0)$ with colors $\alpha_k, \ldots, \alpha_1$, respectively. This yields a $(\Delta + 1)$-edge-coloring of G. $\square$

The above theorem does not hold for graphs with multiple edges. See Figure 9.4.

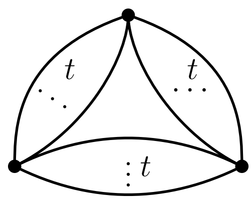


Figure 9.4: $\Delta(G) = 2t, \chi_1(G) = 3t = \Delta(G) + t$.

The following is the generalization of the above theorem for multigraphs.

Theorem 9.2 (V. G. Vizing (1965), R. P. Gupta (1966))**.** *For any loopless graph G,*

$$\chi_1(G) \leq \Delta_1(G) + \mu(G),$$

where $\mu(G)$ denotes the maximum number of edges joining a pair of vertices in G.

9.3 Class-1 and Class-2 graphs

Theorem 9.1 divides the class of simple graphs into two classes.

Definition. *A simple graph G is said to be of **Class-1** if $\chi_1(G) = \Delta(G)$ and it is said to be of **Class-2** if $\chi_1(G) = \Delta(G) + 1$.*

The problem of finding necessary and sufficient conditions for a graph to be of Class-1 (or equivalently Class-2) is a hard unsolved problem. It is called the ***Classification problem***. In the following, we state and prove a few necessary conditions and a few sufficient conditions for a graph to be of Class-1 or Class-2. An alert reader may have noticed that we made similar remarks on the characterization of Hamilton graphs (Chapter 6). It is known that Class-2 graphs are rare. More precisely, it has been shown that if we randomly pick a graph G on n vertices, then the probability that G is Class-1, tends to 1 as $n \to \infty$. For example, among the 143 non-isomorphic simple graphs on at most six vertices only 8 belong to Class-2. See Figure 9.5

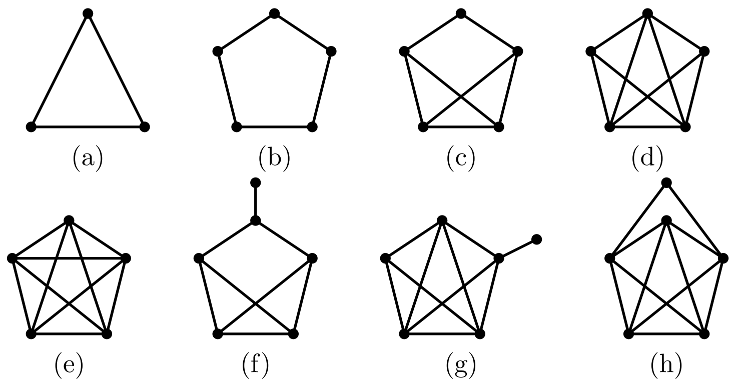


Figure 9.5: All the Class-2 graphs with at most six vertices.

• Edge-coloring of bipartite graphs

The best known theorem on Class-1 graphs is the following.

Theorem 9.3 (König, 1916)**.** *For any bipartite graph G,*

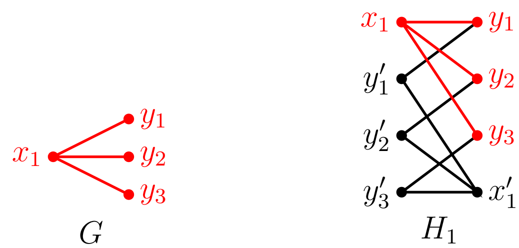
$$\chi_1(G) = \Delta(G).$$

We give two proofs. At the outset, observe that it is enough to prove the theorem for connected graphs, since $\chi_1(G) = \max\{\chi_1(D) : D \text{ is a component of } G\}$.

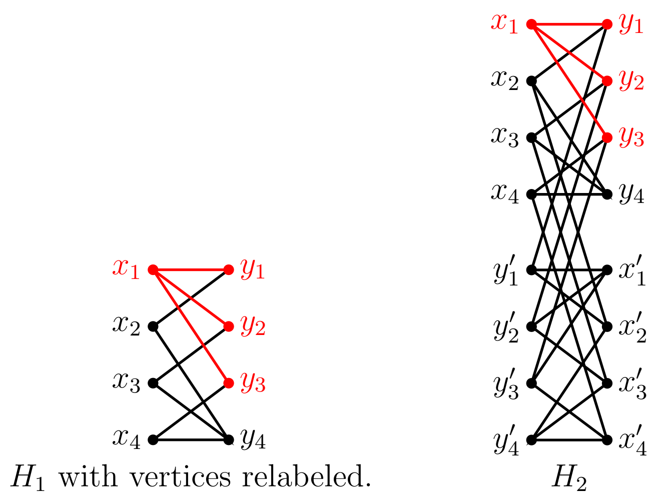
Proof (1). We first construct a $\Delta(G)$-regular bipartite graph H containing G as an induced subgraph. Let $[X, Y]$ be the bipartition of $V(G)$, where $X = \{x_1, x_2, \ldots, x_r\}$ and $Y = \{y_1, y_2, \ldots, y_s\}$. Let $G^1[X^1, Y^1]$ be an isomorphic copy of G, where $X^1 = \{x'_1, x'_2, \ldots, x'_r\}$ and $Y^1 = \{y'_1, y'_2, \ldots, y'_s\}$; see Figure 9.6a. Construct the bipartite graph $H_1[X \cup Y^1, X^1 \cup Y]$ by adding the following edges:

- ○ Join x_i and x'_i $(1 \le i \le r)$, if $\deg_G(x_i) < \Delta(G)$.

- ∘ Join y_i and y'_i $(1 \le i \le s)$, if $\deg_G(y_i) < \Delta(G)$.



(a) $G \to H_1$



(b) $H_1 \to H_2$

Figure 9.6: Construction of a 3-regular bipartite graph which contains G as an induced subgraph.

Then $\Delta(H_1) = \Delta(G)$ and $\delta(H_1) = \delta(G) + 1$. If $\delta(G) + 1 = \Delta(G)$, then H_1 is a $\Delta(G)$-regular bipartite graph with G as an induced subgraph. If $\delta(G)+1 < \Delta(G)$, we repeat the above process with H_1 to construct a bipartite H_2 which contains H_1 (and hence G) as an induced subgraph such that $\Delta(H_2) = \Delta(G)$ and $\delta(H_2) = \delta(H_1) + 1 = \delta(G)+2$. If $\delta(G)+2 = \Delta(G)$, then H_2 is a required graph. We can continue the process enough number of times and construct a sequence of bipartite graphs $H_1, H_2, \ldots, H_t$

such that $G \sqsubseteq H_i \sqsubseteq H_{i+1}$, $\delta(H_{i+1}) = \delta(H_i) + 1$ $(1 \le i \le t-1)$, $\Delta(H_i) = \Delta(G)$ $(1 \le i \le t)$ and $\delta(H_t) = \Delta(G)$. Then H_t is a required bipartite graph.

By Corollary to Hall's Theorem 7.3, $E(H_t)$ can be partitioned into $\Delta(G)$ perfect matchings $E_1, E_2, \ldots, E_{\Delta(G)}$. Therefore, $\chi_1(H_t) \le \Delta(G)$. Since $G \sqsubseteq H_t$, we have $\chi_1(G) \le \chi_1(H_t) \le \Delta(G) \le \chi_1(G)$ and so $\chi_1(G) = \Delta(G)$. □

Proof (2). This proof employs the technique which we used to prove Gupta-Vizing theorem. Assume the contrary and let G be a bipartite graph having minimum number of edges among all the graphs with $\chi_1(G) > \Delta(G)$. So, $\chi_1(G) = \Delta(G) + 1$ but $\chi_1(G-e) = \Delta(G-e) \le \Delta(G)$ $(= \Delta$, say). Let C be a Δ-edge-coloring of $G - e$, where e is (u, v). Let α be a color missing at u and β be a color missing at v; see Figure 9.7. If $\alpha = \beta$, then we can color e with β and get a Δ-coloring of G, a contradiction to our assumption that $\chi_1(G) = \Delta + 1$. Next assume that $\alpha \ne \beta$.

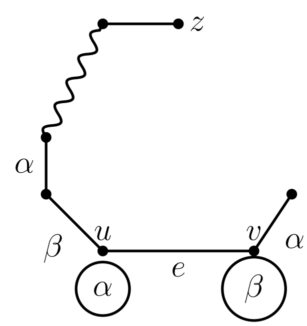


Figure 9.7

There is a β-edge incident with u; else color e with β to get Δ-edge-coloring of G. Similarly, there is a α-edge incident with v.

Let P be a maximal path with origin u and whose edges are colored alternately β and α. Suppose P terminates at a vertex z.

Case 1: $z \ne v$.

Interchange the colors β and α of the edges in P. After this interchange, there is no β-edge incident with u (and with v). Recolor e with β to get a Δ-edge-coloring and thereby a contradiction.

Case 2: $z = v$.

In this case, the first edge of P is a β-edge and the last edge is a α-edge. Therefore, P is a path of even length. Hence, $(u, P(u,v), (v,u))$ is a cycle of odd length in G, which is a contradiction, since G is bipartite. Therefore this case does not arise. □

• Class-2 graphs

In this subsection, we derive a few sufficient conditions for a graph to be of Class-2.

Theorem 9.4. *For any graph G,*

$$\chi_1(G) \geq \left\lceil \frac{m(G)}{\alpha_1(G)} \right\rceil$$

Proof. Let $C = (M_1, \ldots, M_{\chi_1})$ be a χ_1-edge-coloring of G. Then

$$\begin{aligned} m(G) &= |M_1| + \cdots + |M_{\chi_1}|, \\ &\leq \alpha_1(G) + \cdots + \alpha_1(G), \text{ since each } M_i \text{ is a matching in } G, \\ &= \chi_1(G) \cdot \alpha_1(G). \end{aligned}$$

Therefore, $\chi_1(G) \geq \frac{m(G)}{\alpha_1(G)}$. Since χ_1 is an integer, the theorem follows. □

Corollary. *If G is a simple graph with $m(G) > \Delta(G) \cdot \alpha_1(G)$, then G is a Class-2 graph.*

Proof.

$$\begin{aligned}\chi_1(G) &\geq \left\lceil \frac{m(G)}{\alpha_1(G)} \right\rceil, \text{ by Theorem 9.4} \\ &> \left\lceil \frac{\Delta(G)\cdot\alpha_1(G)}{\alpha_1(G)} \right\rceil, \text{ by the hypothesis,} \\ &= \Delta(G).\end{aligned}$$

Therefore, by Gupta-Vizing theorem, $\chi_1(G) = \Delta(G) + 1$. □

Corollary. *If $m(G) > \Delta(G)\lceil \frac{n}{2} \rceil$, then G is Class-2.*

Proof. Exercise. □

Before stating the next result we define a concept which is useful in many other contexts.

Definitions.

- *If v is a vertex in G, then the integer $\Delta(G) - deg(v)$ is called the* ***deficiency*** *of v.*

- *The integer $\sum_{v\in V(G)} (\Delta(G) - deg(v))$ is called the* ***total deficiency*** *of G. We denote it by* ***td(G)****.*

Corollary. *If G is simple, $n(G)$ is odd and $td(G) < \Delta(G)$, then G is Class-2.*

Proof.

$$\Delta(G) \quad > \quad td(G) = \sum_{v\in V(G)} (\Delta(G) - deg(v)) = n\Delta(G) - 2m(G).$$

Therefore,

$$\begin{aligned} m(G) &> \left(\frac{n-1}{2}\right)\Delta(G), \\ &\geq \alpha_1(G)\Delta(G), \text{ since } n \text{ is odd, we have } \alpha_1(G) \leq \frac{n-1}{2}. \end{aligned}$$

Hence, the result follows by the corollary to Theorem 9.4. □

Corollary. *Let $m(G) > 0$. If G is a regular simple graph with $n(G)$ odd, then G is Class-2.*

Proof. $td(G) = \sum_{v \in G}(\Delta(G) - deg(v)) = 0 < \Delta(G)$. Therefore, the result follows by the above Corollary. □

Corollary. *For any $n \geq 2$,*

$$\chi_1(K_n) = \begin{cases} n, & \text{if } n \text{ is odd} \\ n-1, & \text{if } n \text{ is even} \end{cases}$$

Proof. If n is odd, the result follows by the previous corollary. So next assume that n is even. Consider the graph $K_n - v\,(= K_{n-1})$, where v is a vertex of K_n. $\chi_1(K_n - v) = n - 1$, since $n - 1$ is odd. Let $C = (M_1, M_2, \ldots, M_{n-1})$ be a $(n-1)$-edge-coloring of $K_n - v$. Since, $|M_i| \leq \alpha_1(K_n - v) \leq \frac{n-2}{2}$, for every $i, 1 \leq i \leq n-1$, we have $\frac{(n-1)(n-2)}{2} = m(K_n - v) = \sum_{i=1}^{n-1} |M_i| \leq \frac{(n-1)(n-2)}{2}$. Hence, $|M_i| = \frac{n-2}{2}$, for every i, $1 \leq i \leq n-1$. Therefore, there are exactly $n-2$ vertices which are the end vertices of edges in M_1. Hence, there is a vertex (say) v_1 in $K_n - v$ which is not the end-vertex of any edge in M_1, that is, color 1 is missing at v_1. Moreover, every color $\in \{2, 3, \ldots, n-1\}$ is present at v_1, since $deg(v_1) = n - 2$. By repeating the same argument with every M_i, we conclude that there are vertices $v_1, v_2, \ldots, v_{n-1}$ in $K_n - v$ such that the color i is absent at v_i (and every color $\in \{1, 2, 3, \ldots, n-1\} - \{i\}$

is present at v_i). The latter assertion implies that v_i's are all distinct and hence $V(K_n - v) = \{v_1, \ldots v_n\}$. Therefore, coloring the edges $(v, v_1), (v, v_2) \ldots, (v, v_{n-1})$ with colors $1, 2, \ldots, n-1$ respectively, we obtain a $(n-1)$-edge-coloring of K_n. Hence, $\chi_1(K_n) \leq n-1$. Since $\chi_1(K_n) \geq \Delta(K_n) = n-1$, we conclude that $\chi_1(K_n) = n-1$. $\square$

• Hajos union and Class-2 graphs (Optional)

Hajos union can be used to construct a larger Class-2 graph by combining two smaller Class-2 graphs.

Definition (Hajos union). *Let G_1 and G_2 be any two graphs with $\Delta(G_1) = \Delta(G_2) = \Delta$, $\delta(G_1) > 0$ and $\delta(G_2) > 0$. Suppose there exists vertices $x \in V(G_1)$ and $y \in V(G_2)$ such that $deg_{G_1}(u) + deg_{G_2}(v) \leq \Delta + 2$, combine the graphs G_1 and G_2 by applying the following steps:*

1. *Choose any two vertices $u \in V(G_1)$ and $v \in V(G_2)$ such that $deg_{G_1}(u) + deg_{G_2}(v) \leq \Delta + 2$.*

2. *Delete an edge (u, w_1) from G_1 and an edge (v, w_2) from G_2, chosen arbitrarily.*

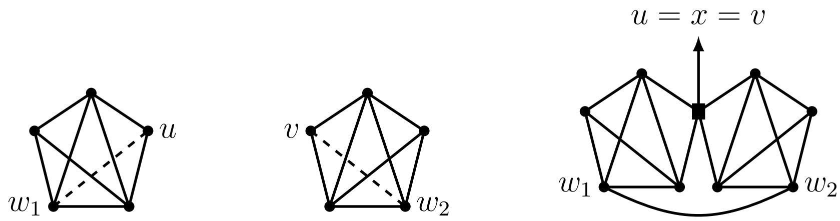
3. *Identify u and v and call the resultant vertex as x.*

4. *Join the vertices w_1 and w_2 by an edge.*

*The resultant graph is called a **Hajos union** of graphs and it is denoted by $G_1 \cup_H G_2$. See Figure 9.8.*

Note that the Hajos union is not a unique graph, since the choice of u, v, (u, w_1) and (v, w_2) are arbitrary.

Theorem 9.5. *Let G_1 and G_2 be Class-2 graphs satisfying the assumptions made in the definition of Hajos union. Then their Hajos union $G = G_1 \cup_H G_2$ is a Class-2 graph.*



(a) G_1 (b) G_2 (c) $G_1 \cup_H G_2$

Figure 9.8: Hajos union of two graphs.

Proof. (Contradiction method) At the outset observe that, for any vertex z in $V(G)$

$$d_G(z) = \begin{cases} d_{G_1}(u) + d_{G_2}(v) - 2, & \text{if } z = x = u = v, \\ d_{G_1}(z), & \text{if } z \neq x \text{ and } z \in V(G_1), \\ d_{G_2}(z), & \text{if } z \neq x \text{ and } z \in V(G_2). \end{cases}$$

Therefore, we conclude that $\Delta(G) = \Delta$.

Next assume that the theorem is false and let $C : E(G) \to \{1, 2, \ldots, \Delta\}$ be a Δ-edge-coloring of G. Without loss of generality let $C(w_1, w_2) = 1$.

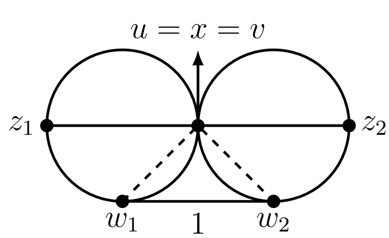


Figure 9.9

Claim: There is a 1-edge, say (u, z_1) in G_1, where $z_1 \in V(G_1)$. Else, by defining $C' : E(G) \to \{1, 2, \ldots, \Delta\}$ by

$$C'(e) = \begin{cases} C(e) & \text{if } e \neq (u, w_1), \\ 1 & \text{if } e = (u, w_1), \end{cases}$$

we obtain a Δ-edge-coloring of G_1, a contradiction. Hence, the claim holds.

Similarly, there is a 1-edge (v, z_2) in G_2, where $z_2 \in V(G_2)$. This implies that there are two 1-edges incident with x, a contradiction to the coloring C. □

9.4 A scheduling problem and equitable edge-coloring (Optional)

An example of a scheduling problem: A conference is planned with five invited speakers S_1, S_2, S_3, S_4, S_5 to handle six short courses $C_1, C_2, C_3, C_4, C_5, C_6$. For the convenience of the participants, on any day a speaker speaks at most once and a course is scheduled at most once.

The following matrix $M = [m_{ij}]_{5\times 6}$ describes the lecture requirements, where m_{ij} is the number of times speaker S_i speaks on the course C_j.

	C_1	C_2	C_3	C_4	C_5	C_6
S_1	0	0	1	0	3	0
S_2	0	2	0	1	0	1
S_3	2	0	0	0	2	0
S_4	0	1	0	2	0	2
S_5	2	0	2	0	0	0

(a) Find the smallest number of days needed to conduct the conference.

(b) Is it possible to schedule the time table where there are only four conference halls available?

Graph theory model and a solution.

Associate the bipartite graph $G[S, C]$ shown in Figure 9.10 with the given matrix M, where S_i and C_j are joined with m_{ij} edges.

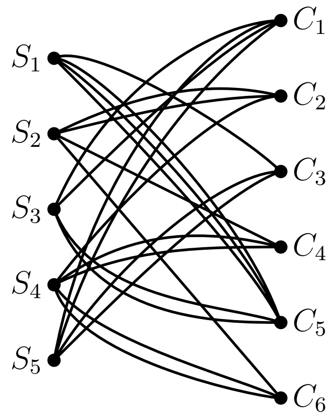


Figure 9.10: The bipartite graph G associated with M.

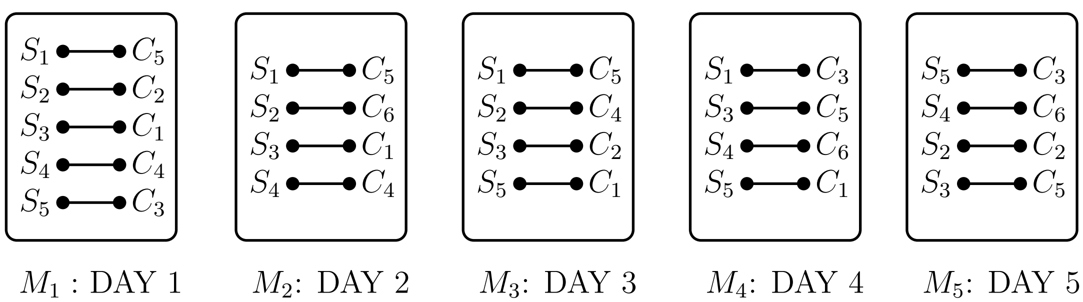


Figure 9.11: Five disjoint matchings of G and the resultant schedule.

Because of the restrictions on the scheduling of speakers and courses, the problem (a), is to partition the edge set of G into subsets $M_1, M_2, \ldots, M_t$ such that each M_i is a matching and t is minimum possible. Obviously, M_i is the schedule for the i-th day. Hence, in the terminology of edge-coloring, we have to find $\chi_1(G)$. Thanks to König's Theorem 9.3, we immediately have that $\chi_1(G) = \Delta(G) = 5$. So, the answer to problem (a) is that five days are required to conduct the conference. An actual schedule is shown in Figure 9.11. And a conference time table derived using this schedule is shown in Table 9.2.

	Course 1	Course 2	Course 3	Course 4	Course 5	Course 6
Day 1	S_3	S_2	S_5	S_4	S_1	–
Day 2	S_3	–	–	S_4	S_1	S_2
Day 3	S_5	S_4	–	S_2	S_1	–
Day 4	S_5	–	S_1	–	S_3	S_4
Day 5	–	S_2	S_5	–	S_3	S_4

Table 9.2: Conference time table derived using the schedule given in Figure 9.11.

We next solve problem (b).

Since there are twenty one edges and there are 5 sets of matchings, at least one matching, say M_1 contains 5 or more edges, by the Pigeon-hole-principle. Since an edge in M_i indicates the requirement of a conference hall on i-th day, 5 conference halls are necessarily required. So, the answer to problem (b) is NO.

If there are only four conference halls available we require six days to conduct the conference. It is easy to modify the table so that every day we require at most four conference halls. For example, we can shift the edge (S_5, C_3) which appears in M_1 to a new matching M_6. But the resultant time table is obviously unbalanced: only one talk on sixth day. So to have a balance, we would like to have a 6-edge-coloring of G such that $||M_i| - |M_j|| \leq 1$, $1 \leq i, j \leq 6$, that is, the number of talks are "nearly equal" on all the days. Next theorem asserts that it is always possible to schedule the time table in a balanced manner.

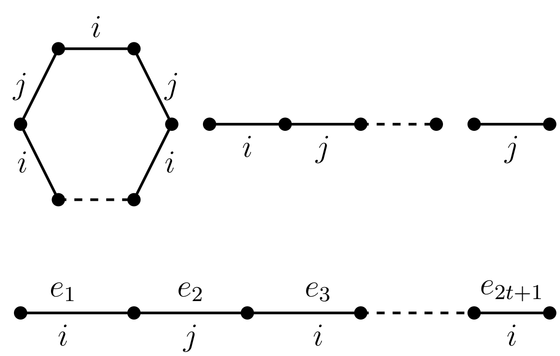
Definition. *A k-edge-coloring $C = (M_1, M_2, \ldots, M_k)$ is said to be a* ***equitable** k**-edge-coloring*** *if $||M_i| - |M_j|| \leq 1$, for every $i, j \in \{1, 2, \ldots, k\}$.*

Theorem 9.6. *If there exists a k-edge-coloring of G, then there exists an equitable k-edge-coloring of G.*

Proof. Let $C = (M_1, M_2, \ldots, M_k)$ be a k-edge-coloring of G. Suppose there exist i and j such that $|M_i| \geq |M_j| + 2$.

We describe a procedure to get matchings M_i' and M_j' such that $|M_i'| = |M_i| - 1$, $|M_j'| = |M_j| + 1$ and $M_i \cup M_j = M_i' \cup M_j'$. By repeating such a procedure enough times, we can get an equitable k-edge-coloring of G.

Consider the subgraph H of G induced by the set of edges $M_i \cup M_j$. $\Delta(H) \leq 2$, since there can be at most one i-edge and at most one j-edge incident with any vertex in H. So, every component of H is either a cycle or a path whose edges are alternately colored i and j. Therefore, any cycle component is an even cycle; see Figure 9.12. We conclude that there exists a path component P in H whose first edge and last edge are colored i, since $|M_i| \geq |M_j| + 2$.



A special path P

Figure 9.12: Components of H.

Define

$$\begin{aligned} M_i' &= M_i - \{e_1, e_3, \ldots, e_{2t+1}\} \cup \{e_2, e_4, \ldots, e_{2t}\}, \\ M_j' &= M_j - \{e_2, e_4, \ldots, e_{2t}\} \cup \{e_1, e_3, \ldots, e_{2t+1}\}. \end{aligned}$$

That is, interchange the colors of P (and retain the colors of edges not in P). Then $M_i \cup M_j = M_i' \cup M_j'$, $|M_i'| = |M_i| - 1$, and $|M_j'| = |M_j| + 1$. $\square$

Exercises

1. Prove:

   (a) If G is a loopless graph with n odd and at least $n \cdot \Delta(G)$ edges, then $\chi_1(G) = \Delta(G) + 1$.

   (b) If G is simple, Hamilton and 3-regular, then $\chi_1(G) = 3$.

2. What is the number of iterations required to construct the regular graph H in the first proof of Theorem 9.3.

3. Let G be a graph with $\chi_1(G) = \Delta(G) + 1$. Let $e(u, v)$ be an edge such that $\chi_1(G - e) = \Delta(G)$. Show that $deg_G(u) + deg_G(v) \geq \Delta(G) + 2$.

4. Find the edge-chromatic number of the following graphs.

   (a) $T + T$, where T is a tree on $n(\geq 3)$ vertices.

   (b) $T_k + C_{2k+1}$, where T_k is a tree on k vertices.

   (c) $K_5 - e$

   (d) $C_5 + C_5$.

   (e) $C_{2s} + C_{2t}$.

   (f) Petersen graph.

   (g) d-cube, Q_d.

   (h) A k-regular bipartite graph.

   (i) $K_{5,5,5,5,5}$.

5. Show that the graphs shown in Figure 9.5 are of Class-2.

6. Let G be a d-regular simple graph with odd number of vertices, where $d \geq 2$. Let H be any graph obtained from G by deleting at most $(d-1)/2$ edges. Show that $\chi_1(H) = \Delta(H) + 1$.

7. Show that every regular simple graph on odd number of vertices is a Class-2 graph.

8. Show that $\chi_1(G \square K_2) = \Delta(G \square K_2)$, for every simple graph G.

9. Prove or disprove: If G_1 and G_2 are Class-1 graphs, then $G_1 + G_2$ is a Class-1 graph.

10. In a school there are seven vacancies, one each in the departments of Chemistry, English, French, Geograph, History, Mathematics and Physics. There are seven applicants $A_1, \ldots, A_7$ for the vacancies. The applicants and their specializations are listed in the table below. Determine the maximum number of suitably qualified teachers the school can employ. Justify your claim.

Teachers	Specialization
A_1	Mathematics, Physics
A_2	Chemistry, English, Mathematics
A_3	Chemistry, French, History, Physics
A_4	English, French, History, Physics
A_5	Chemistry, Mathematics
A_6	Mathematics, Physics
A_7	English, Geography, History.

11. A mathematics department plans to offer seven courses in the next semester namely, Complex Analysis (C), Numerical Methods (N), Linear Algebra (L), Probability and Statistics (P), Differential Equations (D), Special Functions (S) and Graph Theory (G). Each of the following combination of courses has attracted the students.

C,L,D	C,N,G	N,P	C,L
L,P	C, N	S,P	L,N
C,D	C,G,D	S,G	S, D

(a) Use graph theory techniques to find the minimum number of slots required for these courses on a day.

(b) If one wants to schedule the timetable with minimum number of slots, what is the minimum number of class rooms required.

12. There are 5 teachers and 6 classes. The following matrix $M = [m_{ij}]$ describes the teaching requirements in a week, where m_{ij} is the number of times a teacher i meets the class j.

$$M = \begin{bmatrix} 0 & 0 & 3 & 0 & 4 & 0 \\ 0 & 2 & 0 & 2 & 0 & 2 \\ 3 & 0 & 0 & 0 & 4 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 3 & 0 & 2 \end{bmatrix}$$

(a) Find the smalles number of periods needed so that each teacher can meet all the required classes.

(b) Is it possible to schedule the time table where there are only 4 classrooms available?

Module 10 Planar Graphs

Contents

Planar graphs are a major link between graph theory and geometry/-topology. There are three easily identifiable milestones in planar graph theory.

(1) A formula of Euler that $V - E + F = 2$, for any convex polyhedron with V vertices/corners, E edges and F faces.

(2) A deep characterization of planar graphs due to Kuratowski.

(3) The 4-color-theorem of Appel, Haken and Koch.

Colorings of planar graphs made their first appearance in a problem of map-coloring. Recent applications of planar graphs in the design of chips and VLSI have further boosted the current research on planar graphs.

10.1 Basic concepts

In this chapter, we will be guided more by intuition rather than precise definitions. For example the terms, plane, open region, closed region, boundary, interior, exterior of a region in the Euclidean space are not defined. These can be found in any undergraduate text on calculus.

Definition. *A graph G is said to be* ***planar*** *or* ***embeddable in the plane*** *if it can be drawn in the plane so that no two edges intersect except (possibly) at their end vertices; otherwise it is said to be a* ***nonplanar graph****.*

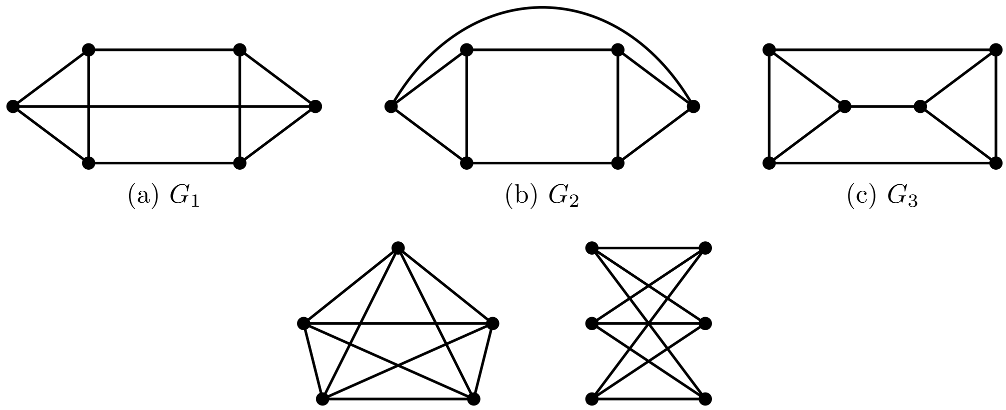
A planar graph embedded in the plane is called a ***plane graph***.

Figure 10.1 shows a planar graph G_1, two plane graphs G_2, $G_3(\simeq G_1)$ and two nonplanar graphs G_4, G_5 (why are they nonplanar?). We emphasize that G_1 is not a plane graph.

Since there are planar graphs and nonplanar graphs, the following three problems arise:

1. Find necessary and sufficient conditions for a graph to be planar.

2. How to test a given graph for planarity?



(a) G_1 (b) G_2 (c) G_3

(d) G_4 (e) G_5

Figure 10.1: Planar graph, plane graphs and nonplanar graphs.

3. Design a (polynomial time) algorithm to draw a given planar graph as a plane graph.

The first problem was solved by Kuratowski in 1930. His characterization uses the hereditary nature of planar graphs. A graph theoretic property P is said to be ***hereditary*** if a graph has property P then all its subgraphs too have property P. Clearly, acyclicity, bipartiteness and planarity are hereditary properties. Kuratowski's characterization has lead to the design of many "good" (:= polynomial time) algorithms to check whether a given graph is planar, and if it is planar to draw it as a plane graph.

A mature reader would have realized that in drawing a planar graph as a plane graph, one would require the knowledge of the famous Jordan Curve Theorem (in the x-y plane).

Given any two points a and b in the plane, any non-self-intersecting continuous curve from a to b is called a ***Jordan curve*** and it is denoted by $J[a, b]$. If $a = b$, then J is called a ***closed Jordan curve***.

Theorem 10.1 (Jordan Curve Theorem)**.**

(i) Any closed Jordan curve J partitions the plane into 3 parts namely, interior of J (int$J$), exterior of J (ext$J$) and J.

(ii) If J is a closed Jordan curve, $s \in$ int$J$ and $t \in$ ext$J$, then any Jordan curve $J'[s, t]$ contains a point of J (that is, J' intersects J).

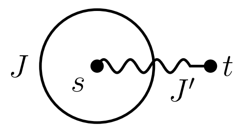


Figure 10.2: Illustration for Jordan Curve Theorem.

If G is a plane graph, then any path in G is identified with a Jordan curve. Similarly, any cycle is identified with a closed Jordan curve. In particular, an edge $e(u, v)$ of G is a Jordan curve from u to v.

Definition. *Let G be a plane graph.*

- *G partitions the plane into several regions. These regions are called the* ***faces*** *of G. The set of all faces of G is denoted by $F(G)$, and the number of faces by $r(G)$; so $|F(G)| = r(G)$.*
- *Except one face, every other face is a bounded region. The exceptional face is called the* ***exterior face*** *and other faces are called* ***interior faces*** *of G. The exterior face is unbounded and interior faces are bounded (:= area is finite).*
- *The* ***boundary*** *of a face f is the set of all edges of G which are incident with f. It is denoted by $b_G(f)$ or $b(f)$.*

Importantly, the boundary of a face f need not be a cycle; it can be a walk; see Figure 10.3.

Definition. *The* ***degree of a face*** *f in a plane graph G is the number of edges in the boundary of f with cut-edges counted twice (the reason why we count twice will be clear soon). The degree of f is denoted by $deg_G(f)$ or $deg(f)$ or $d(f)$.*

The plane graph of Figure 10.3 contains 4 faces.

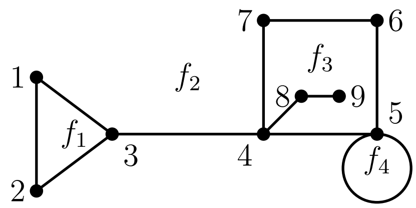


Figure 10.3: A plane graph.

$b(f_1) = \{(1,2),(2,3),(1,3)\}$; $d(f_1) = 3$; $b(f_1)$ is a cycle.

$b(f_2) = \{(1,2),(2,3),(1,3),(3,4),(4,5),(5,5),(5,6),(6,7),(7,4)\}$; $d(f_2) = 10$. Note that (3,4) is counted twice and that $b(f_2)$ does not form a cycle.

$b(f_3) = \{(4,5),(5,6),(6,7),(7,4),(4,8),(8,9)\}$; $d(f_3) = 8$. Note that (4,8), (8,9) are counted twice.

$b(f_4) = \{(5,5)\}$; $d(f_4) = 1$.

Remarks (Consequences of Jordan Curve Theorem)**.**

- A cyclic edge belongs to two faces.
- A cut-edge belongs to only one face.
- A plane graph G is acyclic if and only if $r(G) = 1$.

Theorem 10.2. *If G is a plane graph, then*

$$\sum_{f \in F(G)} deg(f) = 2m.$$

Proof. Every cyclic edge contributes 2 to the left hand side sum since it belongs to 2 faces of G (by Jordan Curve Theorem). Every cut-edge also contributes 2, since it is counted twice although it belongs to only one face. □

Many results on plane graphs become apparent if we look at their "duals".

Definition (Dual of a plane graph). *Let G be a plane graph with $V(G) = \{v_1, v_2, \ldots, v_n\}$, $E(G) = \{e_1, e_2, \ldots, e_m\}$ and $F(G) = \{f_1, f_2, \ldots, f_r\}$. The dual (general) graph G^* of G has vertices $f_1^*, f_2^*, \ldots, f_r^*$ ($f_i \leftrightarrow f_i^*$) and edges $e_1^*, e_2^*, \ldots, e_m^*$ ($e_i \leftrightarrow e_i^*$), where an edge e_j^* joins the vertices f_s^* and f_t^* if and only if the edge e_j is common to the faces f_s and f_t in G.*

The following figure illustrates a drawing of G^*.

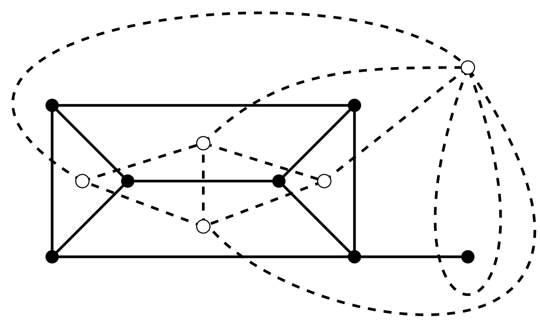


Figure 10.4: A graph G and its dual.

In this drawing, a point is chosen in the interior of every region and two such points f_1^*, f_2^* are joined by a line e^* crossing the edge e only, where e is an edge common to the faces f_1, f_2.

Though the following statements are intuitively obvious from the definition of dual and its drawing, the proofs are tedious and hence they are omitted.

(i) G^* is plane; in fact the above drawing gives a plane embedding of G^*.

(ii) $(G^*)^* = G$.

(iii) e is a cut-edge in G if and only if e^* is a loop in G^*.

Remark. We have defined the dual of a plane graph and not of a planar graph.

10.2 Euler's formula and its consequences

As with many Euler's formulae (like $e^{i\pi} = -1$) his formula for plane graphs too relates three basic parameters and it is both attractive and has many applications.

Theorem 10.3 (Euler, 1758)**.** *For any connected plane graph G,*

$$n - m + r = 2.$$

In polyhedral geometry, the formula is stated using symbols: $V - E + F = 2$.

Proof. (Induction on r) If $r = 1$, then G has no cycles and so it is a tree. Hence $m = n - 1$ and the result holds. Next assume that $r \geq 2$ and that the result holds for all connected plane graphs with $r - 1$ faces. Let G be a plane graph with r faces. Since, $r \geq 2$, G contains a cycle. Let e be a cyclic edge. The graph $G - e$ is a connected plane graph with $n(G - e) = n(G)$, $m(G - e) = m(G) - 1$ and $r(G - e) = r(G) - 1$. The first two equations are obvious and the last equation follows, since e is common to two faces in G and these two faces merge to become one face in $G - e$. By induction hypothesis,

$$n(G - e) - m(G - e) + r(G - e) = 2.$$

So,

$$n(G) - (m(G) - 1) + (r(G) - 1) = 2$$

that is,

$$n(G) - m(G) + r(G) = 2.$$

$\square$

Remark. Euler's formula does not hold for disconnected graphs. Draw a disconnected simple plane graph for which Euler's formula fails.

Corollary. *If G is a connected plane simple graph such that $deg(f) \geq k$, for every face f, then*

$$m \leq \frac{k(n-2)}{k-2}.$$

Proof. Using Theorem 10.2, we find that

$$2m = \sum_{f \in F(G)} deg(f) \geq k \cdot r.$$

Substituting $2m \geq kr$ in Euler's formula, we obtain the result. □

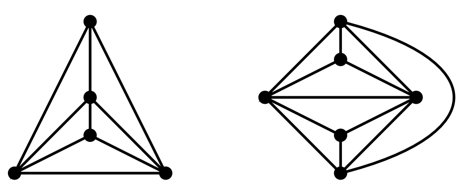
The above result holds for disconnected graphs too.

Corollary. *If G is a plane simple graph such that $deg(f) \geq k$, for every face f, then*

$$m \leq \frac{k(n-2)}{k-2}.$$

□

Definition. A simple planar graph G is said to be a maximal planar graph if $G + e$ is nonplanar, for every $e \in G^c$. Two maximal plane graphs are shown below.



(a) G_1 (b) G_2

Figure 10.5: Maximal plane graphs.

Remarks.

- By definition, every maximal planar graph is simple.

- Every simple planar graph is a spanning subgraph of a maximal planar graph. (Go on adding edges until addition of any edge results in a nonplanar graph.)
- Every face of a maximal plane graph is a triangle. Hence a maximal plane graph is called a ***triangulation of the sphere***.
- Every maximal planar graph ($n \geq 3$) is 2-vertex-connected.

Corollary. *If G is a planar simple graph on at least 3 vertices, then $m \leq 3n - 6$.*

Proof. It is enough if we prove the result for maximal plane graphs; see the remarks above. So, assume that G is a maximal plane graph on at least three vertices. For every face f in G, $deg(f) \geq 3$. So, by putting $k = 3$ in the previous corollary, we find that

$$m \leq \frac{3(n-2)}{3-2} = 3n - 6.$$

□

Corollary. *K_5 is nonplanar.*

Proof. On the contrary, if K_5 is a plane graph, then using the above corollary, we get $10 = m(K_5) \leq 9$, a contradiction. □

Corollary. *$K_{3,3}$ is nonplanar.*

Proof. Assume the contrary that $K_{3,3}$ is a plane graph. Since, it is a bipartite graph, $deg(f) \geq 4$, for every face f. Therefore by putting $k = 4$, in the first corollary to Euler's theorem, we get

$$9 = m(K_{3,3}) \leq \frac{4(4)}{2} = 8,$$

a contradiction. □

Corollary. *If G is a plane simple graph, then $\delta(G) \leq 5$.*

Proof. On the contrary, if $\delta(G) \geq 6$, then $2m = \sum_{v \in V(G)} deg(v) \geq 6n$ which is a contradiction. □

Corollary. *If G is a connected plane simple graph with $\delta(G) \geq 3$, then it contains a face of degree ≤ 5.*

Proof. Assume the contrary that every face f in G has degree ≥ 6. Then

(i) $2m = \sum_{f \in F(G)} deg(f) \geq 6r$.
Since, $\delta(G) \geq 3$, we also have

(ii) $2m = \sum_{v \in V(G)} deg(v) \geq 3n$.

Using these two inequalities in Euler's formula, we get

$$2 = n - m + r \leq \frac{2m}{3} - m + \frac{m}{3} = 0,$$

a contradiction. □

10.3 Polyhedrons and planar graphs (Optional)

Polyhedrons have been fascinating objects since ancient times. In the following, all the polyhedrons are convex.

That the graph of a polyhedron is planar can be practically demonstrated as follows:

Take a polyhedron (P) made of elastic sheet. Color the nodes and the sides of P. Thereupon make a hole in one of the faces of P. Stretch P so that P becomes a plane sheet with the circle of the hole as the boundary. Get a color print of the nodes and the sides of P by pressing it on a paper. The color print is precisely the graph of the polyhedron and it is plane.

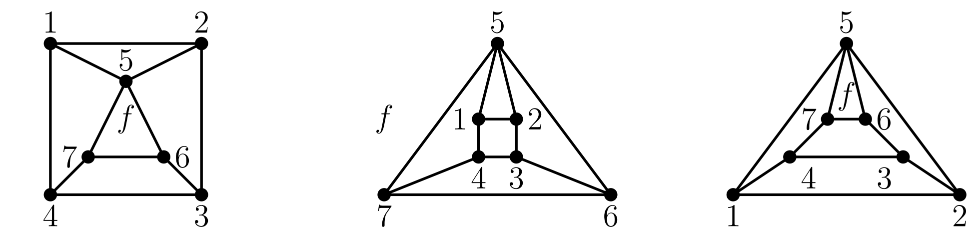
The above operation can be described in precise mathematical formulation known as ***stereographic projection***.

The stereographic projection of a plane graph implies a few crucial facts.

(1) If G is a plane graph and f is any face in G then G can be embedded in the plane so that f is the exterior face.

(2) If G is a plane graph and v is any vertex, then G can be embedded in the plane such that v lies on the exterior face. The same conclusion holds for any edge of G.

A few illustrations are given below.



(a) A plane graph G (b) G_1 (c) G_2

Figure 10.6: G_1 is a redrawing of G such that the interior face f of G is the exterior face of G_1. G_2 is a redrawing of G such that the edge (1,5) lies on the exterior face in G_2. Both are drawn using suitable stereographic projections as explained in the above remark.

The above statements are useful to prove the following result.

Theorem 10.4.

1. *A graph G is planar iff every component of G is planar.*
2. *A graph G is planar iff every block of G is planar.*

Proof. (1) is obvious.

(2) If G is planar, then every block of G is planar, since every subgraph of a planar graph is planar.

Conversely, suppose every block of G is planar. We show that G is planar by induction on the number of blocks $b(G)$ in G. If $b(G) = 1$, there is nothing to be proved. Next suppose that $b(G) \geq 2$. So G contains a cut-vertex say x. Hence, there exist subgraphs G_1 and G_2 of G such that $V(G_1) \cup V(G_2) = V(G)$, $V(G_1) \cap V(G_2) = \{x\}$, $E(G_1) \cup E(G_2) = E(G)$, $E(G_1) \cap E(G_2) = \emptyset$. Since $b(G_1) < b(G)$ and $b(G_2) < b(G)$, G_1 and G_2 are planar by induction hypothesis. By one of the remarks above, G_1 and G_2 can be embedded in the plane such that x belongs to the exterior face of G_1 and also to the exterior face of G_2; see Figure 10.7. Then G_1 and G_2 can be combined to get a plane drawing of G. □

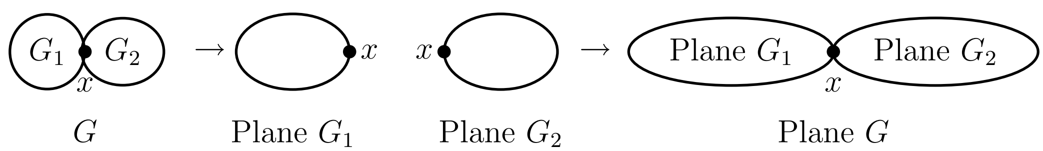


Figure 10.7: A plane drawing of G using plane drawing of G_1 and G_2.

Definitions.

1. *A polyhedron is called a* ***regular polyhedron*** *if*

   (i) *the boundaries of faces are congruent polygons,*

   (ii) *equal number of faces surround each corner.*

2. *The polyhedral graph $G(P)$ of a polyhedron P consists of corners of P as its vertices and sides of P as its edges (:= the 1-skeleton of P).*

It is known that $G(P)$ is a 3-vertex-connected plane simple graph and conversely, if H is a 3-vertex-connected plane simple graph, then there exists a convex polyhedron P such that $G(P) = H$.

In Figure 10.8, three polyhedrons are shown (their graphs are apparent). The first and the third polyhedron are regular polyhedrons and the second polyhedron is not a regular polyhedron.

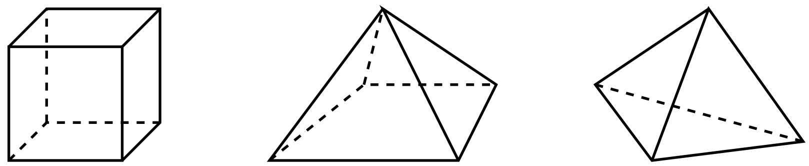
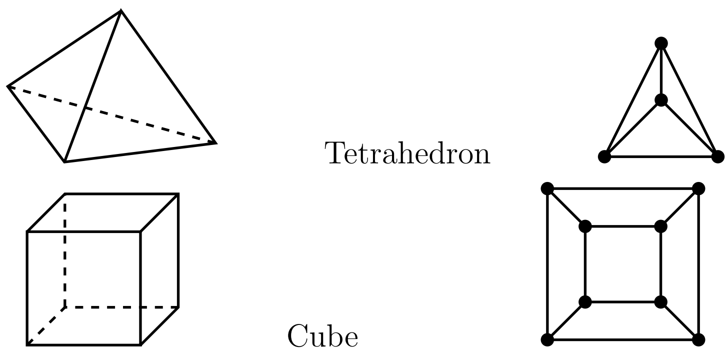


Figure 10.8: Three polyhedrons.

In Figure 10.9, on the left we have 5 regular polyhedra (known as ***Platonic solids***) and on the right their associated plane graphs. Surprisingly, these are the only five regular polyhedra. We can use Euler's formula to prove this statement.



Theorem 10.5. *There are exactly five regular polyhedra.*

Proof. Let P be a regular polyhedron with V nodes, E sides and F faces. Since, the nodes, sides and faces of P are precisely the vertices, edges and faces of $G(P)$, we have

(i) $V - E + F = 2$,

(ii) $\sum\limits_{face\, f \in P} deg(f) = 2E$,

(iii) $\sum\limits_{node\, v \in P} deg(v) = 2E$.

Since P is regular, every node has degree k (say) and every face has degree p (say). Then (ii) and (iii) reduce to $pF = 2E$ and $kV = 2E$, respectively. So,

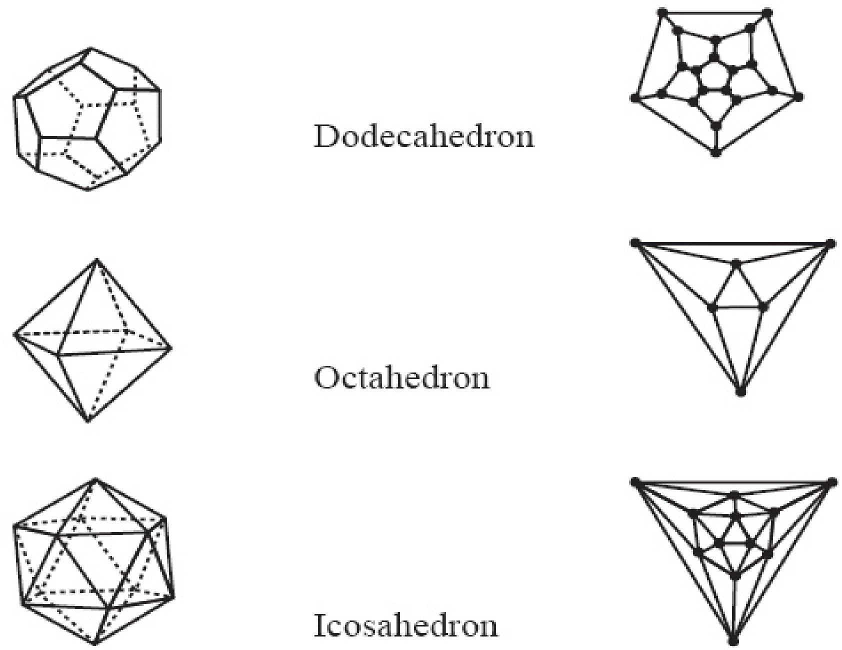


Figure 10.9: Platonic solids and their plane graphs.

(iv) $kV = pF = 2E$.
The equations (i) and (iv) imply that

(v) $(k-4)V + (p-4)F = -8$.

Clearly, every face of a polyhedral graph is bounded by at least 3 edges, and so the minimum degree of faces is 3. Hence, by the corollaries from previous section, we have $3 \leq k \leq 5$ and $3 \leq p \leq 5$. Thus there are 9 possible solutions (k, p) of (v).

Case 1. $(k, p) = (3, 3)$.

By (iv) and (v), we get $V = F = 4$. Hence P is a regular polyhedron with 4 nodes, 4 faces with every node and every face of degree 3. So, P is the tetrahedron.

Case 2. $(k, p) = (3, 4)$.

From (iv) and (v), we deduce that $V = 8$ and $F = 6$. Thus P is a regular polyhedron, with 8 vertices and 6 faces, in which every node has degree 3 and every face has degree 4. Hence, P is the cube.

Case 3. $(k, p) = (3, 5)$.

In this case (iv) and (v) yield $V = 20$ and $F = 12$. Thus P is the dodecahedron.

Case 4. $(k, p) = (4, 3)$.

By (iv) and (v), we have $V = 6$ and $F = 8$. Hence, P is the octahedron.

Case 5. $(k, p) = (4, 4)$.

With $k = 4$ and $p = 4$, (v) reduces to $0 = -8$, which is absurd. Hence (4,4) is not a solution of (v).

Case 6. $(k, p) = (4, 5)$.

In this case, (v) reduces to $F = -8$; and obviously there is no polyhedron with $F = -8$.

Case 7. $(k, p) = (5, 3)$.

Using (iv) and (v) we get $V = 12$ and $F = 20$. Hence, P is the icosahedron.

Case 8. $(k, p) = (5, 4)$.

From (v), we get $V = -8$; and clearly there is no polyhedron with $V = -8$.

Case 9. $(k, p) = (5, 5)$.

In this case (v) reduces to $V + F = -8$; and clearly there is no polyhedron with $V + F = -8$. □

10.4 Characterizations of planar graphs

There are four major characterization of planar graphs, due to Kuratowski (1930), Whitney (1932), Wagner (1937) and MacLane (1937). We state and prove Kuratowski's theorem, and state Wagner's theorem.

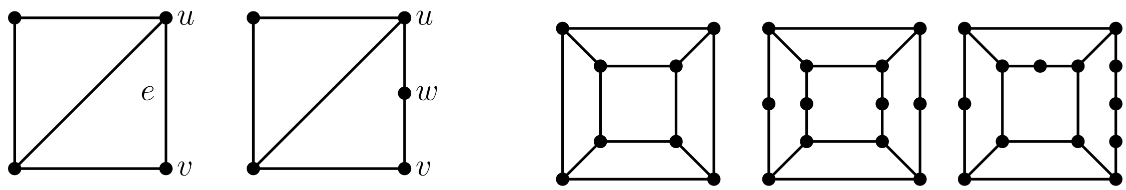
• Subdivisions and Kuratowski's characterization

Central to Kuratowski's characterization is the concept of 'subdivision'.

Definitions.

- *The **subdivision** of an edge $e(u, v) \in E(G)$ is the operation of replacing e by a path (u, w, v), where w is a new vertex. So, to get a subdivision of e introduce a new vertex w on e.*
- *A graph H is said to be a **subdivision of** G if H can be obtained from G by a sequence of edge subdivisions. (By definition, G is a subdivision of G.)*

See Figure 10.10.



(a) A subdivision of e (b) Two subdivisions of a cube

Figure 10.10: Examples of subdivision.

Notation: Any subdivision of G is denoted by $S(G)$. Note that $S(G)$ is not unique. In fact, there are infinite number of subdivisions of any graph with at least one edge.

Remarks.

(1) $H \subseteq G$, G is planar $\Rightarrow H$ is planar.

(2) $H \subseteq G$, H is nonplanar $\Rightarrow G$ is nonplanar.

(3) G is planar $\Rightarrow S(G)$ is planar.

(4) G is nonplanar $\Rightarrow S(G)$ is nonplanar.

(5) $S(K_5)$ and $S(K_{3,3})$ are nonplanar. (A consequence of corollaries from Section 10.2 and (4).)

(6) G is planar $\Rightarrow G \not\supseteq S(K_5), S(K_{3,3})$. (A consequence of (2) and (5))

A natural question is to ask whether the converse of (6) holds. That is, $G \not\supseteq S(K_5), S(K_{3,3}) \Rightarrow G$ is planar? Kuratowski's theorem asserts that it indeed holds.

The Figure 10.11 illustrates that the Petersen graph is nonplanar.

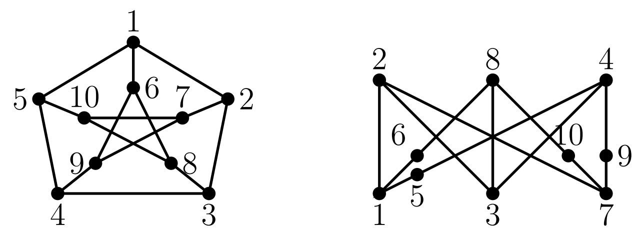


Figure 10.11: The Petersen graph contains a subdivision of $K_{3,3}$. Therefore, the Petersen graph is nonplanar by Remarks (5) and (2) above.

There have been several attempts to simplify and strengthen the original proof of Kuratowski's theorem. Here, we follow Dirac and Schuster (1954) with an excellent terminology of Bondy and Murty (1976).

It is obvious that G is planar if and only if its underlying simple graph is planar. So in the following we assume that all our graphs are simple.

Theorem 10.6 (Kuratowski's characterization of planar graphs). *A graph G is planar if and only if G contains no $S(K_5)$, $S(K_{3,3})$.*

Proof. We have remarked before that if G is planar, then G contains no $S(K_5)$, $S(K_{3,3})$.

To prove the converse, we assume the contrary that there exists a graph which contains no subdivision of K_5 or $K_{3,3}$ but it is nonplanar. Among all such graphs, let G be a graph with ***minimum number*** of edges. In view of Theorem 10.4, $k_0(G) \geq 2$. We consider two cases: $k_0(G) = 2$ and $k_0(G) \geq 3$.

Case 1: $k_0(G) = 2$.

Let $\{u, v\}$ be a vertex-cut of G. There exist two connected graphs G_1 and G_2 of G such that $V(G_1) \cup V(G_2) = V(G)$, $V(G_1) \cap V(G_2) = \{u, v\}$ and $E(G_1) \cup E(G_2) = E(G)$. For $i = 1, 2$, define

$$H_i = \begin{cases} G_i, & \text{if } (u,v) \in E(G) \\ G_i + (u,v), & \text{if } (u,v) \notin E(G). \end{cases}$$

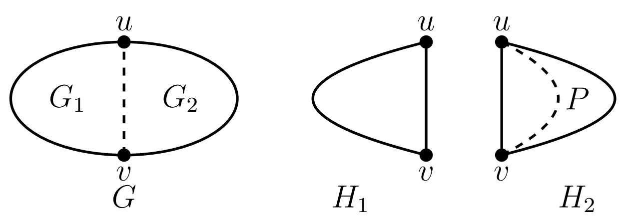


Figure 10.12: Decomposition of a graph into parts.

Claim: $H_1 \not\supseteq S(K_5), S(K_{3,3})$.

If $H_1 = G_1 (\subseteq G)$, the claim is obvious. Next suppose that $H_1 = G_1 + (u, v)$ and that $H_1 \supseteq S(K_5)$ (or $S(K_{3,3})$-proof in this case is similar). Necessarily, $(u, v) \in S(K_5)$. Consider the graph $S(K_5) - (u, v) + P(u, v)$, where $P(u, v)$ is a path with all its internal vertices from G_2; see Figure 10.12. Then $S(K_5) - (u, v) + P(u, v)$ is a subdivision of K_5, and it is a subgraph of G, a contradiction. Therefore, the claim holds.

Similarly, $H_2 \not\supseteq S(K_5)$, $S(K_{3,3})$. Therefore, H_1 and H_2 are planar graphs by the minimality of $E(G)$. We can embed both the graphs H_1 and H_2 in the plane such that the edge (u, v) lies in the exterior face. But then we can combine H_1 and H_2 to get a plane embedding of G, a contradiction to our assumption that G is nonplanar.

Case 2: $k_0(G) \geq 3$.

Let $e(u,v)$ be any edge in G and consider the graph $G-e$. It is planar by the minimality of $m(G)$. Assume that we are given a plane embedding of $G-e$. Since $k_0(G) \geq 3$, we have $k_0(G-e) \geq 2$ (see Exercise 45 from Chapter 2). Therefore, there exists a cycle containing u and v (; see the corollary to Theorem 2.7). Among all such cycles, we choose a cycle C which contains maximum number of edges in the $extC$. For discussion sake, we fix clockwise direction for C. If $x, y \in V(C)$, we write $x \leq y$, if x precedes y on C. The symbol $C[x,y]$ denotes the set of vertices $\{v \in V(C) : x \leq v \leq y\}$. Similarly, $C\ (x,y]$, $C(x,y)$ and $C\ [x,y)$ are defined.

Let D be a component of $G-e-V(C)$. We adopt the following terminology.

- The subgraph B of $G-e$ consisting of D and the set of edges which join a vertex of D with a vertex of C is called a ***branch*** of C. A vertex of C which is joined to a vertex of D is called a ***contact vertex*** of B. If B has k contact vertices, then it is called k***-branch***.
- In addition, an edge (x,y) of $G-e$ where x and y are two non-consecutive vertices of C is also called a branch; x and y are called its ***contact vertices***. To emphasize its nature, we also refer to such a branch as an ***edge-branch***.

Figure 10.13 shows various branches.

We make a series of observations which imply that either G is planar or $G \not\supseteq S(K_5)$ or $G \not\supseteq S(K_{3,3})$ (thus arriving at a contradiction).

(1) Some branches lie in the $extC$ and some branches lie in the $intC$. These are respectively called ***outer-branches*** and ***inner-branches***.

(2) Every branch is a k-branch with $k \geq 3$ or it is an edge-branch, since $k_0(G) \geq 3$.

(3) Every inner-branch is an edge-branch.
On the contrary, suppose H is an inner-k-branch with $k \geq 3$. Then at least two of its contact vertices lie in $C[u,v]$ (or $C[v,u]$). But then there exists a cycle C' in $G-e$ containing u, v which contains more number of edges in the $extC'$ than C - a contradiction to the choice of C; see Figure 10.14a.

(4) Every inner-edge-branch has a contact vertex in $C(u,v)$ and a contact vertex in $C(v,u)$; else, we get a cycle C' as described in (3); see Figure 10.14b.

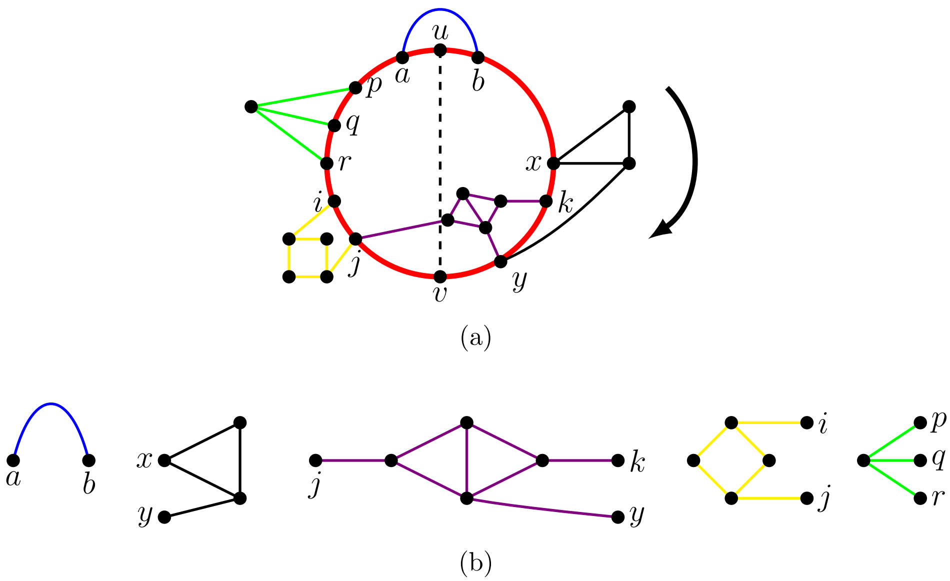


Figure 10.13: (a) A cycle C in $G-e$ with four outer-branches and two inner-branches. (b) Branches of C. (Outer-branches and inner branches are defined below.)

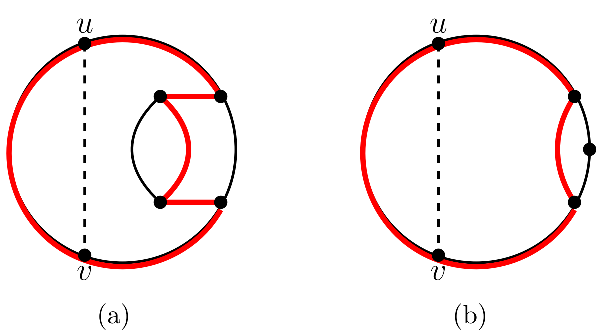


Figure 10.14: Steps in the proof of assertions (3) and (4).

(5) There exists an inner branch, say (r, s) where $r \in C(u, v)$ and $s \in C(v, u)$; else e can be drawn in the $intC$ and get a plane embedding of G, a contradiction. See Figure 10.15.

(6) There exists an outer-branch B with a contact vertex (say, a) in $C(u, v)$ and a contact vertex (say, b) in $C(v, u)$; else e can be drawn in the $extC$ and get a plane embedding of G, a contradiction.

So we have:

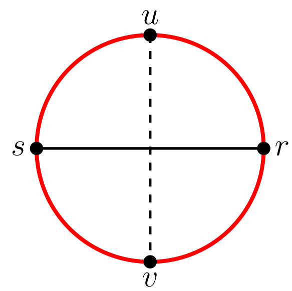


Figure 10.15

(i) $a \in C(u,v) = C(u,r] \cup C[r,v)$,

(ii) $b \in C(v,u) = C(v,s] \cup C[s,u)$.
Because of symmetry it is enough if we deal with the following two cases (other cases follow similarly).

(iii) $a \in C(r,v)$, $b \in C(s,u)$,

(iv) $a = r$, $b = s$.

Suppose (iii) holds. There exists a path $P(a,b)$ in G with all its internal vertices in $V(B) - V(C)$. But then $G \supseteq S(K_{3,3})$ as shown in Figure 10.16, a contradiction.

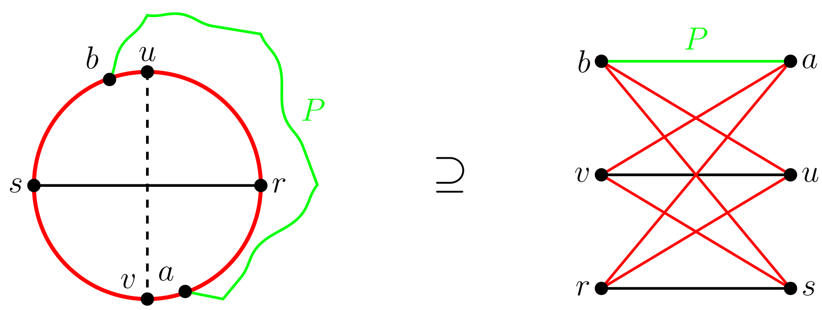


Figure 10.16: $G \supseteq S(K_{3,3})$.

Next suppose (iv) holds, that is $a = r$ and $b = s$.

We next make two observations.

(7) (s,r) cannot be drawn in the *extC* by the maximality of number of edges in the *extC*.

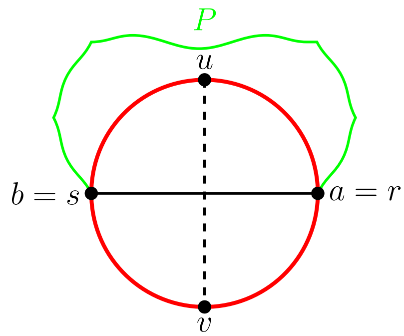


Figure 10.17

(8) No outer-branch ($\neq B$) has a contact vertex in $C(r, s)$ and a contact vertex in $C(s, r)$ (Jordan Curve Theorem).
These two observations imply that:

(9) B has a contact vertex say, $x \in C(s, r)$ and a contact vertex $y \in C(r, s)$. It is possible that $x = u$ or $y = v$. So, we consider two subcases.

Subcase 1: $x = u$ and $y = v$.

There exist paths $P(s, r)$ and $Q(x, y)$ with all their internal vertices from $V(B) - V(C)$; P and Q share at least one common vertex (Jordan Curve Theorem).

If P and Q have exactly one common vertex, say z, then $G \supseteq S(K_5)$, where $V(K_5) = \{u, v, s, r, z\}$; see Figure 10.18.

If P and Q share more than one common vertex, let z_1 and z_2 be the first common vertex and last common vertex, respectively; see Figure 10.19. Then there exist pairwise internally vertex disjoint paths $P(z_1, s), Q(z_1, u), S(z_1, z_2), P(z_2, r)$ and $Q(z_2, v)$. Therefore, $G \supseteq S(K_{3,3})$; see Figure 10.19.

Subcase 2: $\{u, v\} \neq \{x, y\}$; for definiteness, let $v \neq y$. Recall that $y \in C(r, s) = C(r, v) \cup C\,[v, s)$. Because of symmetry, we deal only with the case $y \in C(r, v)$. In view of (iii), we can assume that $x \in C\,[u, r)$; see Figure 10.20 below.

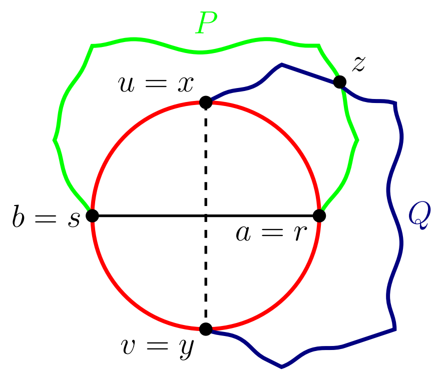


Figure 10.18: Existence of paths $P(s, r)$ and $Q(u, v)$ with a common vertex z and $G \supseteq S(K_5)$.

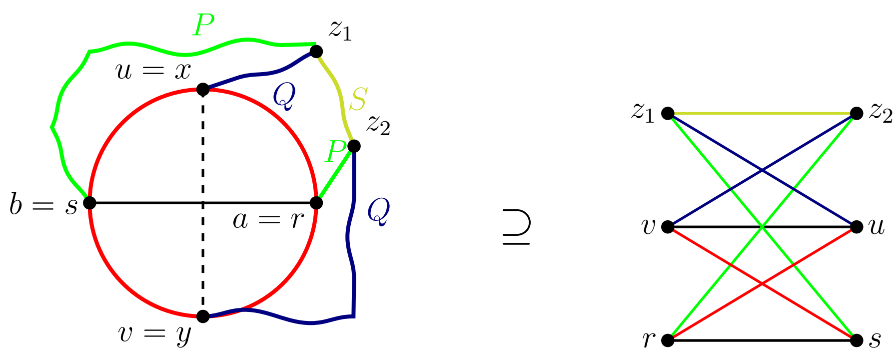


Figure 10.19: $G \supseteq S(K_{3,3})$.

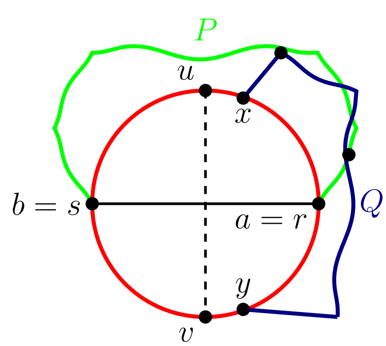
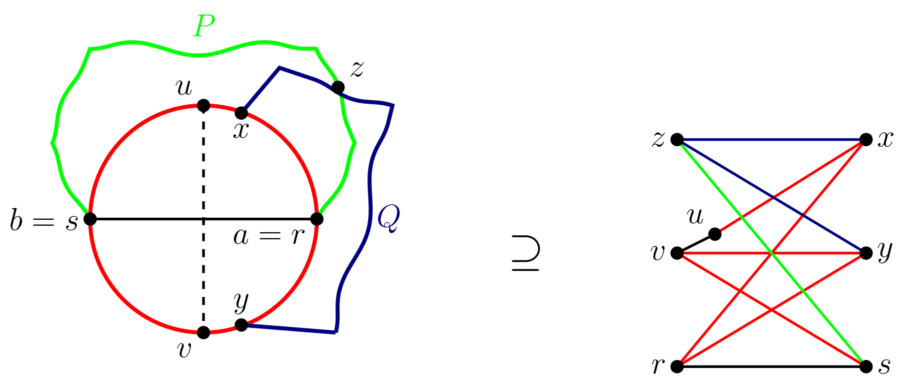


Figure 10.20: Intersection of $P(s, r)$ and $Q(x, y)$.

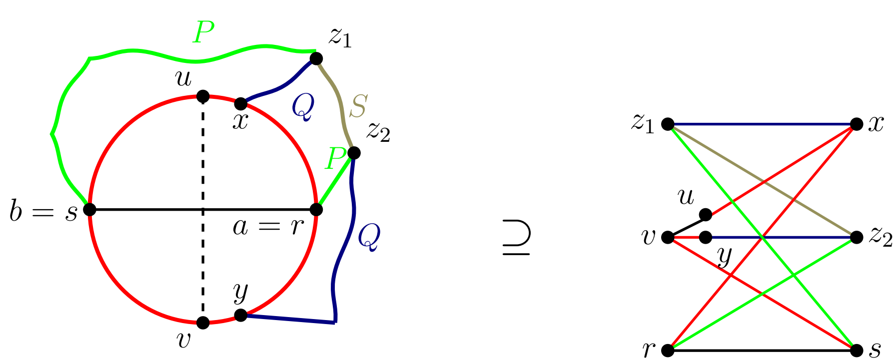
As in the previous case, consider the paths $P(s, r)$ and $Q(x, y)$. They share at least one common vertex. If P and Q have exactly one vertex in common, let it be z. If they have more than one vertex in common, let z_1 be the first common vertex and z_2 be the last common vertex. In both the cases $G \supseteq S(K_{3,3})$; see Figure 10.21.



P and Q intersect in exactly one vertex z.

$S(K_{3,3})$.

(a)



P and Q intersect in more than one vertex.

$S(K_{3,3})$

(b)

Figure 10.21: $G \supseteq S(K_{3,3})$.

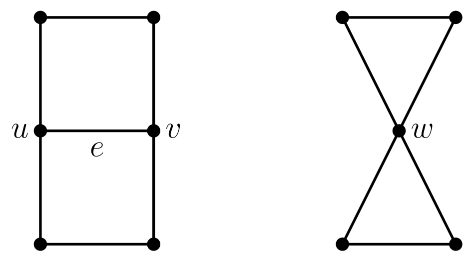
This completes the proof. $\square$

• Minors and Wagner's theorem

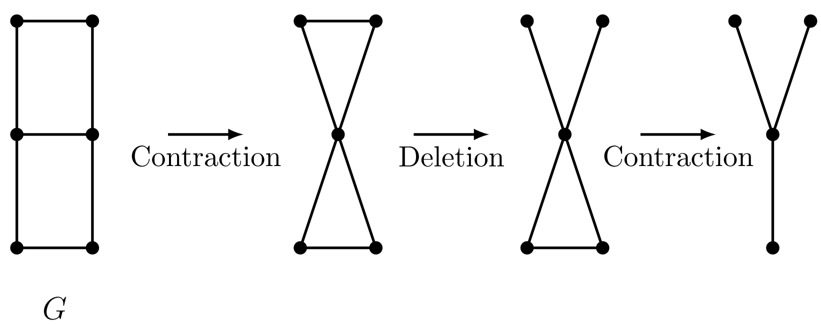
Definitions.

(1) Let $e(u, v)$ be an edge in a graph G. Delete the vertices u and v. Add a new vertex w to $G - \{u, v\}$ and join w to all those vertices in $V - \{u, v\}$ to which u or v is adjacent in G. This operation is called the edge ***edge contraction*** *of e. The resultant graph is denoted by $G \cdot e$.*

(2) A graph H is said to be a ***minor*** *of G, if an isomorphic copy of H can be obtained from G by deleting or contracting a sequence of edges. By convention, G is a minor of G.*



(a) Contraction of e.



(b) $K_{1,3}$ is a minor of G.

Figure 10.22: Edge contraction and minors.

Remarks.

(1) If $G \supseteq S(H)$, then H is a minor of G. H can be obtained by (i) deleting the edges of $E(G) - E(S(H))$ from G, and then (ii) contracting the edges incident to some of the vertices of degree 2.

(2) However, the converse of (1) is false, that is if H is a minor of G, then G need not contain $S(H)$. For example, K_5 is a minor of the Petersen graph but it contains no $S(K_5)$. See Figure 10.23.

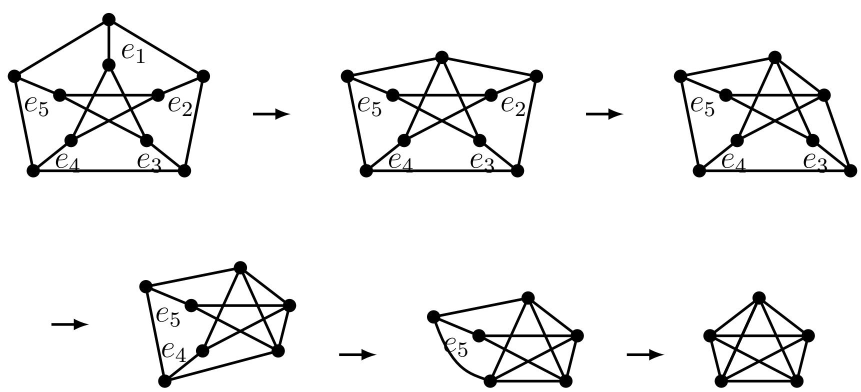


Figure 10.23: K_5 is a minor of the Petersen graph.

Theorem 10.7 (Wagner, 1937)**.** *A graph G is planar if and only if neither K_5 nor $K_{3,3}$ is a minor of G.* □

10.5 Planarity testing (Optional)

Based on Kuratowski's theorem, an algorithm to test the planarity of a graph was designed by Demoucron, Melgrange and Pertuiset (1964). To describe this algorithm, we require the concept of branches of any subgraph $H \subseteq G$. In the proof of Kuratowski's theorem, we dealt with the branches of a cycle in G.

Definitions. *Let H be a subgraph of a graph G. Let D be a component of $G - V(H)$.*

- *The subgraph B of G which consists of D and the set of edges which join a vertex of D with a vertex of H is called a* ***branch*** *of B.*

- *A vertex of H which is joined to a vertex of D is called a **contact vertex** of B.*
- *If B has k contact vertices, then it is called a **k-branch**. The set of all contact vertices of B is denoted by $V(B;H)$.*
- *In addition, an $(x, y) \in E(G) - E(H)$ with $x, y \in V(H)$ is called an **edge-branch**; x and y are its contact vertices.*
- *If $H \subseteq G$ is a plane graph, then a branch B of H is said to be **drawable** in a face f of H if $V(B;H) \subseteq$ boundary of f. The set $\{f \in F(H) : B$ is drawable in $f\}$ is denoted by $F(B;H)$.*

• D-M-P-planarity algorithm

Input: A 2-vertex-connected simple graph G.

Output: A plane embedding of G or a declaration that G is nonplanar.

Step 1: Choose a cycle in G, say G_1, and let $\tilde{G}_1$ be a plane embedding of G_1.

Having drawn a plane embedding $\tilde{G}_i$ of a subgraph $G_i (i \geq 1)$ of G, do the following:

Step 2: If $E(G) - E(\tilde{G}_i) = \emptyset$, then ***stop*** and declare that G is planar and output $\tilde{G}_i$ (; it is a plane drawing of G).

Else, find all the branches of $\tilde{G}_i$. For each such branch B, find $F(B; \tilde{G}_i)$.

Step 3:

(i) If there exists a branch B such that $F(B; \tilde{G}_i) = \emptyset$, ***stop*** and declare that G is nonplanar.

(ii) If there exists a branch B such that $|F(B; \tilde{G}_i)| = 1$, let $f \in F(B; \tilde{G}_i)$.

(iii) If $|F(B; \tilde{G}_i)| \geq 2$, for every branch B of $\tilde{G}_i$ choose any such B and let f be any face $\in F(B; \tilde{G}_i)$.

Step 4: Select a path $P_i \subseteq B$ which connects two contact vertices of B. Draw P_i in f. Set $\tilde{G}_{i+1} = \tilde{G}_i \cup P_i$ and goto step 2 with $\tilde{G}_{i+1}$.

Remark. *Algorithm may fail to correctly test the planarity of G if in Step 4 the entire B is drawn in f instead of the path P_i.*

Illustration 1:

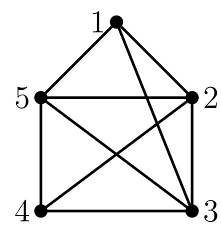
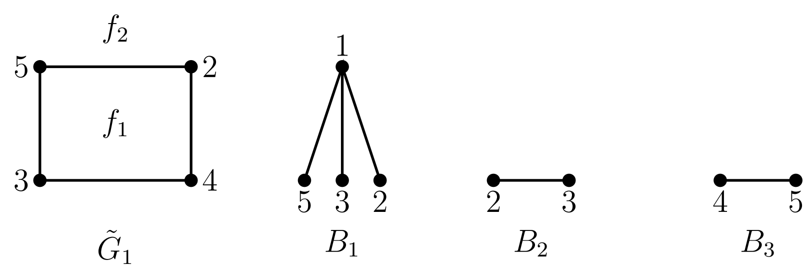


Figure 10.24: Input graph G

Iteration 1: Following Step (1), we arbitrarily choose the cycle $\tilde{G}_1 = (2, 4, 3, 5, 2)$.

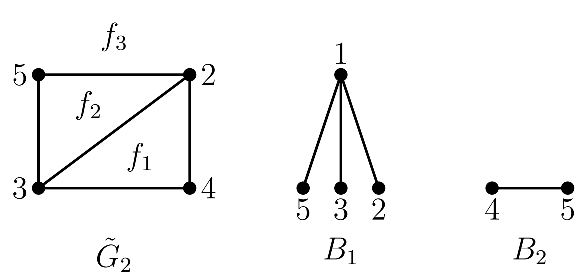


$\tilde{G}_1$ and its branches.

$$F(B_1; \tilde{G}_1) = \{f_1, f_2\},\ F(B_2; \tilde{G}_1) = \{f_1, f_2\} \text{ and } F(B_3; \tilde{G}_1) = \{f_1, f_2\}.$$

Here Step 3(iii) applies. We arbitrarily choose B_2 and f_1 to enlarge the drawing of $\tilde{G}_1$.

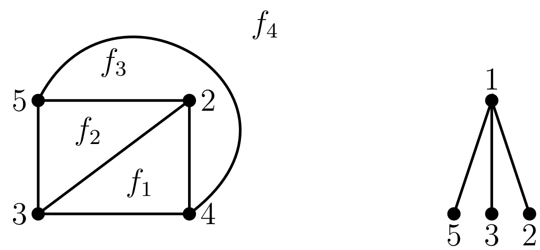
Iteration 2:



$\tilde{G}_2$ and its branches. $F(B_1; \tilde{G}_2) = \{f_2, f_3\}$ and $F(B_2; \tilde{G}_2) = \{f_3\}$.

Here we are compelled to choose B_2 and f_3 by Step 3(ii).

Iteration 3:

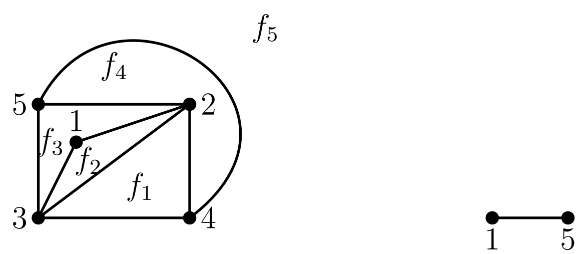


$\tilde{G}_3$ B_1

$\tilde{G}_3$ and its branch B_1. $F(B_1; \tilde{G}_3) = \{f_2\}$.

Again we are compelled to choose B_1 and f_2. Following Step 4, we can either choose the path $(2, 1, 3)$ or $(3, 1, 5)$ or $(2, 1, 5)$; we choose $(2, 1, 3)$.

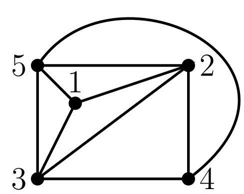
Iteration 4:



$\tilde{G}_4$ B_1

$\tilde{G}_4$ and its branch B_1. $F(B_1; \tilde{G}_4) = \{f_3\}$.

Iteration 5:



$\tilde{G}_5$

At this step, $E(G) - E(\tilde{G}_5) = \emptyset$. So $\tilde{G}_5$ is a plane embedding of G.

Illustration 2:

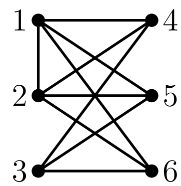
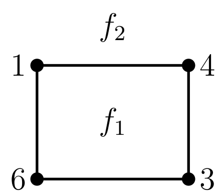
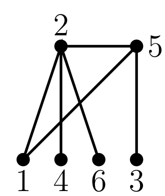


Figure 10.25: Input graph G.

Iteration 1:



$\tilde{G}_1$

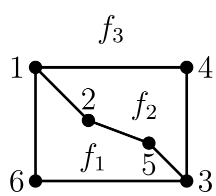


B_1

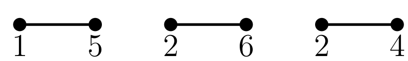
$\tilde{G}_1$ and its branch B_1. $F(B_1; \tilde{G}_1) = \{f_1, f_2\}$.

We choose f_1 and the path $(1, 2, 5, 3) \subseteq B_1$.

Iteration 2:



$\tilde{G}_2$

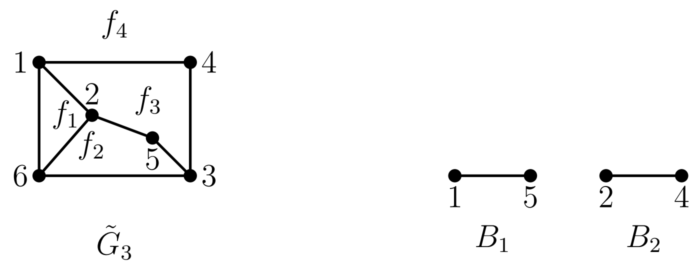


B_1 B_2 B_3

$\tilde{G}_2$ and its branches. $F(B_1; \tilde{G}_2) = \{f_1, f_2\}$, $F(B_2; \tilde{G}_2) = \{f_1\}$ and $F(B_3; \tilde{G}_2) = \{f_2\}$.

We choose B_2 and f_1.

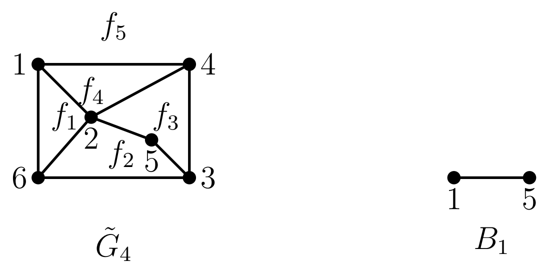
Iteration 3:



$\tilde{G}_3$ and its branches. $F(B_1; \tilde{G}_3) = \{f_3\}$ and $F(B_2; \tilde{G}_3) = \{f_3\}$.

We choose B_2.

Iteration 4:



$\tilde{G}_4$ and its branch $B_1 : F(B_1, \tilde{G}_4) = \emptyset$.

By Step 2 of the algorithm, we conclude that G is nonplanar.

10.6 5-Color-theorem

As remarked in the beginning of this chapter, planar maps and planar graphs first appeared in a problem called the ***four color conjecture*** (1850).

Four-Color-Conjecture
Any map of a country can be colored with at most four colors so that no two adjacent states receive the same color.

It was a fascinating open problem for a long time, which attracted many well-known mathematicians. Their insights, proof techniques, and variations of the problem laid foundation for the topic of graph colorings. The conjecture was finally solved by K. Appel, W. Haken and J. Koch (1977). Their proof techniques involved making of a large number of cases by a computer; thus making the computer necessary in a mathematical proof for the first time in the history of mathematics. The original proof consisted of 700 pages and consumed about 1200 hours of CPU time in 1970's. So, the proof generated a lot of debate and is still continuing. There have been attempts to simplify the proofs but no current proof is less than 100 pages.

Any map M represents a plane graph G whose faces represent the states of M. As observed earlier its dual G^* is also a plane graph. Clearly, the face-coloring of G is equivalent to the vertex-coloring of G^*. So the conjecture can be restated as follows:

Four-Color-Conjecture (Alternative form)

Every plane graph is 4-vertex-colorable.

It is easy to show that every plane graph G is 6-vertex-colorable by induction on n, or by using greedy algorithm since $\delta(G) \leq 5$. However, to show that every plane graph is 5-vertex-colorable we require new ideas.

The first published proof for the 4-vertex-colorability is due to A.B. Kempe (1879). However, an error was found by P.J. Heawood in 1890, and the same author showed in 1898 that Kempe's argument can be used to prove the following weaker form of the Conjecture.

Theorem 10.8 (Heawood, 1898)**.** *Every planar graph is 5-vertex-colorable.*

Proof. (By induction on n). If $n \leq 5$, then obviously the result holds. So, we proceed to the induction step, assuming that every planar graph on $n-1$ vertices is 5-vertex-colorable. Let G be a planar graph on n vertices, and let v be a vertex of minimum degree in G. By a corollary proved earlier, $d(v) \leq 5$. Consider the planar graph $G - v$

and assume that we are given a plane embedding of $G - v$. By induction hypothesis, $G - v$ is 5-vertex-colorable; let $f : V(G) \to \{1, 2, 3, 4, 5\}$ be a 5-vertex-coloring. Our aim is to extend f to a 5-vertex-coloring of G.

If $d(v) \leq 4$, then at most four colors appear in its neighborhood $N(v)$. So, we can color v with one of the five colors which does not appear in $N(v)$.

Next, suppose $d(v) = 5$, and let $N(v) = \{v_1, v_2, v_3, v_4, v_5\}$. See Figure 10.26.

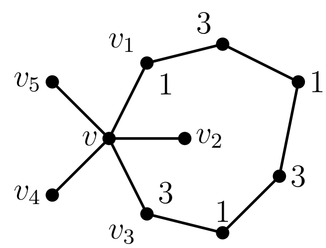


Figure 10.26: Five neighbors of v.

If there appear at most four colors among the colors of v_1, v_2, v_3, v_4, v_5, then the missing color can be used to color v. So, assume that all the five colors appear in the neighborhood of v and that $f(v_i) = i, 1 \leq i \leq 5$.

Let $G_{i,j}$ denote the subgraph of $G - v$ induced by the vertices colored i or j. Consider $G_{1,3}$; $v_1, v_3 \in V(G_{1,3})$.

If v_1 and v_3 are in different components of $G_{1,3}$ say $v_1 \in D_1$ and $v_3 \in D_2$, then we exchange the colors 1 and 3 of vertices in D_1, without disturbing the colors of other vertices. The resultant coloring is a 5-vertex-coloring of $G - v$ such that $f(v_1) = 3, f(v_2) = 2, f(v_3) = 3, f(v_4) = 4$ and $f(v_5) = 5$. So we can color v with 1 and get a 5-vertex-coloring of G.

Next assume that v_1 and v_3 belong to the same component of $G_{1,3}$; so there exists a (v_1, v_3)-path P in $G_{1,3}$. Then $C = (v, v_1, P(v_1, v_3), v_3, v)$ is a cycle in G whose interior contains v_2 and exterior contains v_4 and v_5 or interior contains v_4 and v_5 and exterior contains v_2. Let $v_2 \in intC$ and $v_4 \in extC$; see Figure 10.26. Now consider $G_{2,4}$. If v_2 and v_4 are in different components, then we can get a 5-vertex-coloring of G as before. So, assume that v_2 and v_4 belong to the same component. Therefore, there exists a (v_2, v_4)-path say Q in $G_{2,4}$. Note that the vertices of P are colored alternately 1 and 3, and the vertices of Q are colored alternately 2 and 4.

Since $v_2 \in intC$ and $v_4 \in extC$, P and Q intersect. Since $G - v$ is a plane graph, the intersection point is a vertex, say x of G. Since $x \in V(P)$, it is colored 1 or 3. Since $x \in V(Q)$, it is colored 2 or 4. We thus arrive at a contradiction.

Hence, we conclude that (i) v_1 and v_3 are in different components of $G_{1,3}$, or (ii) v_2 and v_4 are in different components of $G_{2,4}$. In either case, we can extend f to a 5-vertex-coloring of G. $\square$

Exercises

1. Draw plane simple graphs with the following degree sequences:

$$(4^3, 2^3), (5^{12}), (6^2, 5^{12}), (6^3, 5^{12}), (4^3, 3^8), (7, 4^{10}, 3^5).$$

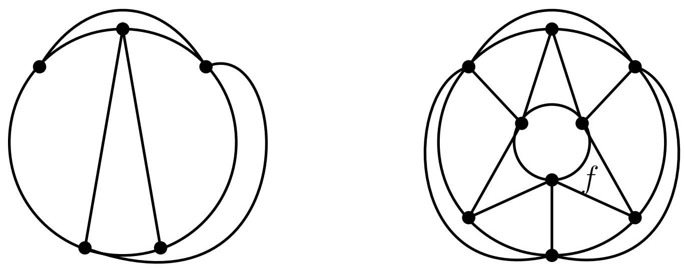
   (The general problem of finding necessary and sufficient conditions for a graphic sequence to realize a planar graph is ***open***.)

2. Redraw the graphs shown in Figure 10.27 so that all the edges are straight lines and no two lines intersect. (It is known that every planar simple graph can be drawn as a plane graph in which every edge is a straight line.)

3. Redraw the graph in Figure 10.27b so that the face f is the exterior face.

4. Use the first corollary of Theorem 10.3 to show that the Petersen graph is nonplanar.

5. Let G be a plane graph with r faces and c components. Show that $n + r = m + c + 1$.



(a) (b)

Figure 10.27

6. Find which of the platonic graphs (Figure 10.9) are Hamilton.

7. Let G be a connected plane 3-regular graph and let n_k denote the number of faces of degree k.

   (a) Show that $\sum_{k\geq 3}(k-6)n_k + 12 = 0$.

   (b) Deduce from (a) that if G is bipartite, then G contains at least 6 faces of degree 4.

8. Find the number of components in a simple plane graph which has 11 vertices, 13 edges and 6 faces.

9. Draw a connected plane simple graph H with 9 edges and 6 faces. Find the edge-chromatic number of the dual of H.

10. If G is a simple planar bipartite graph, then show that $m(G) \leq 2n(G) - 4$.

11. Draw all the non-isomorphic plane simple 3-regular graphs with exactly 5 faces.

12. (a) Show that there exists no 3-regular simple planar graph on 10 vertices whose girth is 5.

    (b) Draw a plane 3-regular simple graph on 10 vertices whose girth is 4.

    (c) Draw a plane 3-regular simple graph on 10 vertices whose girth is 3.

13. Find all the values of k, for which there exists a k-regular simple plane graph.

14. If G is a simple maximal planar graph with $\Delta(G) = 7$, then show that $n_7 = 3n_3 + 2n_4 + n_5 - 12$, where n_k denotes the number of vertices of degree k in G.

15. Prove or disprove: There exists a simple 4-regular maximal plane graph.

16. Find the number of faces in a maximum plane simple graph on 10 vertices.

17. Prove or disprove: There exists a maximal planar simple graph ($n > 1$) whose complement is also a maximal planar.

18. A tree on $n\,(\geq 3)$ vertices is such that its complement is maximum planar. Find n. Draw one such tree.

19. Show that every graph with at most 3 cycles is planar.

20. Find the number of faces in a plane embedding of $K_{1,1,n}$.

21. Find all the values of n such that $K_{1,2,\ldots,n}$ is planar.

22. Find all the values of a, b, c such that $K_{a,b,c}$ is planar, where $1 \leq a \leq b \leq c$.

23. Prove or disprove: If G is a simple planar graph with $n \leq 11$, then G contains a vertex of degree ≤ 4.

24. If G is a nonplanar graph then show that either

    (i) there are at least 5 vertices of degree ≥ 4, or

    (ii) there are at least 6 vertices of degree ≥ 3.

25. Show that in a planar graph G (with $n \geq 4$) there are at least 4 vertices of degree ≤ 5.

26. If G is a simple planar graph with $\delta(G) = 5$, then show that there are at least 12 vertices of degree 5. Give an example of planar graph on 12 vertices with $\delta(G) = 5$.

27. If $n(G) \geq 11$, then show that G or G^c is nonplanar. Give an example of a graph G on 8 vertices such that G and G^c are planar.

28. If G is a planar with degree sequence $(d_1 \geq d_2 \geq \cdots \geq d_n)$, then show that

$$\sum_{i=1}^{k} d_i \leq 6(k-2) + \sum_{i=k+1}^{n} d_i, \text{ for every } k,\ 1 \leq k \leq n.$$

29. Show that the following are equivalent for a connected graph G.

(a) G is a tree.

(b) G contains no subdivision of K_3.

(c) G contains no K_3-minor.

30. Verify whether the graphs shown below are embeddable in the plane by applying D-M-P-algorithm. Draw all the intermediate plane graphs and their branches generated by the algorithm.

(a) $C_5 + K_1$.

(b) $C_5 + K_2$.

(c) $K_{2,2,2,2}$.

(d) The graph with vertices 1, 2, 3, 4, 5, 6, 7, 8 and edges (1,2), (2,3), (3,4), (4, 5), (5, 6), (6, 7), (7, 8), (8,1), (1,5), (2,6), (3,7), (4,8).

31. Use the greedy algorithm to show that every planar graph is 6-vertex-colorable.

32. Prove Kuratowski's theorem (Case 2) by choosing a cycle C in $G - e$ which contains maximum number of edges in the $intC$.

33. Let G be a simple plane connected 3-regular graph with every face having degree 5 or 6. If p is the number of pentagonal faces and h is the number of hexagonal faces in G, show that $p = 12$ and $n(G) = 20 + 2h$.
(Which motivational problem stated in Chapter 1 and section 1 is now solved?)

Module 11 Directed Graphs

Contents

Graphs that we studied in chapters 1 to 10 are inadequate to model many real world problems. These include one-way message routings and Turing machine computations. In all such problems one requires the notion of direction from one node to another node.

11.1 Basic concepts

Definition. *A **directed graph** D is a triple (V, A, I_D) where V and A are disjoint sets and $I_D : A \to V \times V$ is a function.*

- *An element of V is called a **vertex**.*
- *An element of A is called an **arc**.*

Often we call a directed graph as a ***digraph***. As in the case of graphs we assume that V and A are finite sets and denote $|V|$ by n and $|A|$ by m. If more than one graph are under discussion, we denote V, A, n and m by $V(D), A(D), n(D)$ and $m(D)$, respectively.

An example of a digraph:

Let $V = \{1, 2, 3, 4, 5\}$, $A = \{a, b, c, d, e, f, g, h\}$ and $I_D : A \to V \times V$ be defined by $I_D(a) = (1, 2)$, $I_D(b) = (2, 3)$, $I_D(c) = (3, 2)$, $I_D(d) = (4, 3)$, $I_D(e) = (4, 1)$, $I_D(f) = (4, 1)$, $I_D(g) = (3, 5)$, $I_D(h) = (5, 5)$. Then (V, A, I_D) is a digraph with 5 vertices and 8 arcs. It is represented as follows:

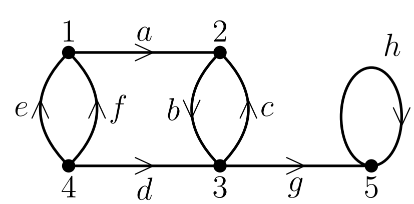


Figure 11.1: A digraph.

As in the case of graphs e and f are multiple arcs and h is a loop. However, b and c are not multiple arcs.

If $I_D(x) = (u, v)$, then x is denoted by $x(u, v)$ and say that

- ◦ u is the ***tail*** of x.
- ◦ v is the ***head*** of x.
- ◦ x is an arc ***from*** u ***to*** v.
- ◦ If $x(u, u)$ is a loop, then u is its head and also its tail.
- ◦ A directed graph with no multiple arcs and no loops is called a ***simple digraph***.

• Underlying graph of a digraph

The underlying graph $G(D)$ of a digraph $D(V, A, I_D)$ is obtained by ignoring the direction of its arcs. Formally, $V(G) = V$, $E(G) = A$ and $I_G : A \to V^{(2)}$ is defined by $I_G(a) = (u, v)$, if $I_D(a) = (u, v)$ or (v, u). The underlying graph of a digraph is shown below.

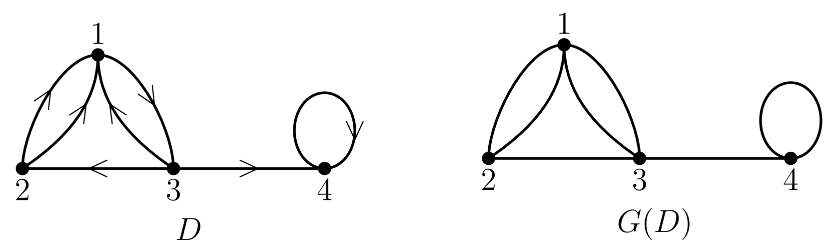


Figure 11.2: A digraph and its underlying graph.

Often we call a graph as an ***undirected graph*** if both a graph and a digraph are under discussion.

The various subdigraphs of a digraph D are defined analogous to subgraphs of a graph. We continue to use the notation $D - W$ (where $W \subseteq V(D)$) and $D - B$

(where $B \subseteq A(D)$) for the subdigraphs obtained from D by deleting a set of vertices and a set of arcs.

The concepts of degrees, walks and connectivity are defined taking into account the direction of arcs.

• Out-degrees and in-degrees

Definitions. *Let D be a digraph and v be a vertex.*

- *The **out-degree** of v is the number of arcs with v as their tail. It is denoted by $outdeg_D(v)$.*
- *The set*
$$N_{out}(v) = \{x \in V(D) : (v, x) \in A(D)\}$$
*is called the **set of out-neighbors** of v. Clearly, if D is **simple**, then $|N_{out}(v)| = outdeg(v)$.*
- *The **in-degree** of v is the number of arcs with v as their heads. It is denoted by $indeg_D(v)$.*
- *The set*
$$N_{in}(v) = \{x \in V(D) : (x, v) \in A(D)\}$$
*is called the **set of in-neighbors** of v. Clearly, if D is simple, then $|N_{in}(v)| = indeg(v)$.*

So each vertex v in a digraph is associated with an ordered pair $(outdeg(v), indeg(v))$ of integers; see Figure 11.3.

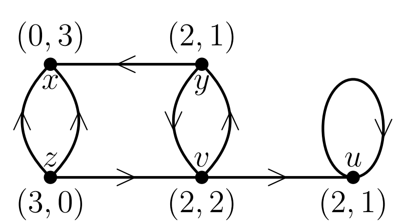


Figure 11.3: Out-degrees and in-degrees of vertices. $N_{out}(x) = \emptyset$, $N_{in}(x) = \{y, z\}$, $N_{out}(y) = \{x, v\}$, $N_{in}(y) = \{v\}$, $N_{out}(u) = \{u\}$, $N_{in}(u) = \{u, v\}$.

Theorem 11.1. *For every digraph D,*

$$\sum_{v \in V(D)} outdeg(v) = \sum_{v \in V(D)} indeg(v) = m.$$

Proof. Every arc is counted once in $\sum outdeg(v)$ and once in $\sum indeg(v)$. $\square$

• Isomorphism

Definition. *Two digraphs $D_1(V_1, A_1, I_{D_1})$ and $D_2(V_2, A_2, I_{D_2})$ are said to be **isomorphic** if there exist bijections $f : V_1 \to V_2$ and $g : A_1 \to A_2$ such that x is an arc from u to v in D_1 if and only if $g(x)$ is an arc from $f(u)$ to $f(v)$ in D_2.*

*The pair of functions (f, g) is called an **Isomorphism**. If D_1 and D_2 are isomorphic, we write $D_1 \simeq D_2$.*

Figure 11.4 shows isomorphic and non-isomorphic digraphs.

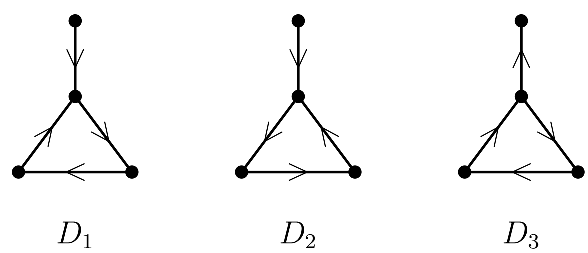


Figure 11.4: $D_1 \simeq D_2, D_1 \not\simeq D_3$.

Remark. If $D_1(V_1, A_1)$ and $D_2(V_2, A_2)$ are simple digraphs, then $D_1 \simeq D_2$ if and only if there exists a bijection $f : V_1 \to V_2$ such that $(u, v) \in A_1$ if and only if $(f(u), f(v)) \in A_2$.

11.2 Directed walks, paths and cycles

Definitions. *Let D be a digraph and let $v_0, v_t \in V(D)$.*

- *An alternating sequence*

$$W(v_0, v_t) := (v_0, a_1, v_1, a_2, v_2, \ldots, v_{t-1}, a_t, v_t)$$

*of vertices and arcs where a_i $(1 \leq i \leq t)$ is an arc from v_{i-1} to v_i is called a (v_0, v_t)-**directed walk** or a **directed walk from** v_0 **to** v_t. Here, the vertices or arcs need not be distinct.*

- *v_0 is called the **origin** and v_t is called the **terminus** of $W(v_0, v_t)$. Its length is defined to be t, the number of arcs, where an arc is counted as many times as it occurs.*

- *A (v_0, v_t)-walk is called a (v_0, v_t)-**trail** if all its arcs are distinct.*

- *A (v_0, v_t)-walk is called a (v_0, v_t)-**path** if all its vertices (and) hence arcs are distinct. A path is denoted by the sequence of vertices alone if no confusions is anticipated.*

- *A $W(v_0, v_t)$ is called a **closed directed walk** if $v_0 = v_t$.*

- *A closed directed walk $W(v_0, v_t)$ is called a **directed cycle** if all its vertices are distinct except that $v_0 = v_t$.*

Remark. In all these definitions, the adjective "directed" can be dropped if it is clear from the context that we are concerned with directed graphs.

We illustrate these concept by taking a digraph.

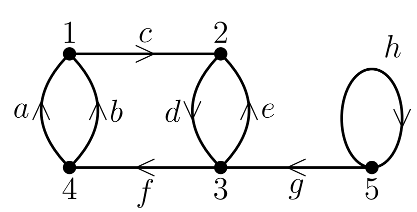


Figure 11.5: A digraph D.

(i) $W_1(5, 4) = (5, g, 3, e, 2, d, 3, e, 2, d, 3, f, 4)$
is a directed walk of length 6. It is neither a trail nor a path.

(ii) $W_2(5,4) = (5, g, 3, e, 2, d, 3, f, 4)$
is a directed trail of length 4. It is not a path.

(iii) $W_2(5,4) = (5, g, 3, f, 4)$
is a directed path of length 2.

(iv) $W_3(1, c, 2, d, 3, f, 4, b, 1)$
is a directed cycle of length 4.

Remarks.

- If there exists a (v_0, v_t)- directed walk, it is not necessary that, there exists a (v_t, v_0)-directed walk. In the above example, we have a (5,4)-directed walk but no (4,5)-directed walk.
- If there exists a (v_0, v_t)-directed walk, there exists a (v_0, v_t)-directed path.

• Connectivity in digraphs

Definitions.

- *A digraph is said to be* ***weakly connected*** *if its underlying graph is connected; otherwise, it is said to be disconnected.*
- *A digraph is said to be* ***unilaterally connected*** *if given any two vertices u and v, there exists a directed path from u to v or a directed path from v to u.*
- *A digraph is said to be* ***strongly connected*** *or* ***strong*** *if given any two vertices u and v, there exists a directed path from u to v and a directed path from v to u.*

Clearly, D is strongly connected $\Rightarrow$ D is unilaterally connected $\Rightarrow$ D is weakly connected. The converse implications do not hold; see Figure 11.6.

Theorem 11.2. *A digraph D is strong if and only if it contains a closed directed walk which contains all its vertices.*

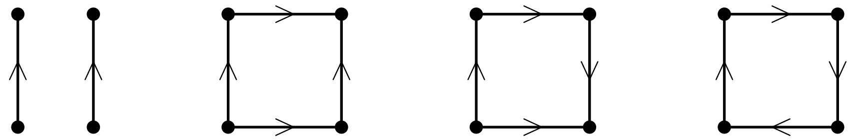


Figure 11.6: D_1 is disconnected; D_2 is weakly connected but it is not unilaterally connected; D_3 is unilaterally connected but it is not strongly connected; D_4 is strongly connected.

Proof. (1) D is strong $\Rightarrow$ D contains a closed directed walk which contains all its vertices.

Since D is strong, if $u, v \in V(D)$, then there exist directed paths $P_1(u, v)$ and $P_2(v, u)$. Therefore, D contains a closed directed walk $(P_1(u, v), P_2(v, u))$. Among all the closed directed walks, let $W(x, x)$ be a closed directed walk containing maximum number of vertices.

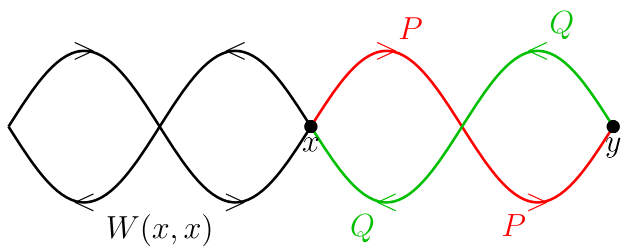


Figure 11.7: Extension of $W(x, x)$.

Our aim is to show that W contains all the vertices of D. On the contrary, suppose that there exists a vertex $y \in V(D) - V(W)$. Since D is strong, there exist directed paths $P(x, y)$ and $Q(x, y)$; see Figure 11.7. But then $(W(x, x), P(x, y), Q(y, x))$ is a closed directed walk containing more number of vertices than W, a contradiction to the maximality of W. Therefore, $W(x, x)$ contains all the vertices of D.

(2) D contains a closed directed walk which contains all the vertices of $D \Rightarrow D$ is strong.

Let $W(x, x)$ be a closed directed walk containing all the vertices of D. Let $u, v \in V(D) = V(W)$. Since W is closed, W contains directed subwalks $W_1(u, v)$ and $W_2(v, u)$. Hence D is strong. $\square$

Theorem 11.3. *A digraph D is unilaterally connected if and only if it contains a directed walk (not necessarily closed) containing all the vertices of D.*

Proof. (1) D is unilateral $\Rightarrow D$ contains a directed walk containing all the vertices of D.

Among all the directed walks in D, let W be a directed walk containing maximum number of vertices of D. Let $W = W(u_0, u_t) = (u_0, e_1, u_1, e_2, u_2, \ldots, u_{t-1}, e_t, u_t)$. We assert that W contains all the vertices of D. On the contrary, suppose that there exists a vertex $x \in V(D) - V(W)$; see Figure 11.8.

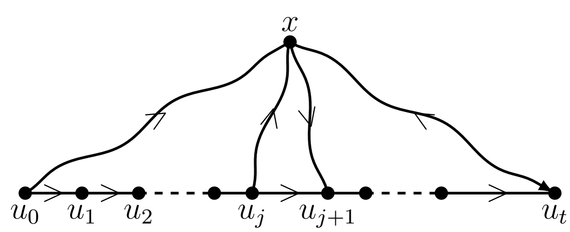


Figure 11.8: Extension of $W(u_0, u_t)$.

Claim 1: There exists a (u_0, x)-walk.

Else, there exists a (x, u_0)-walk, say $W'(x, u_0)$, since D is unilaterally connected. But then $(W'(x, u_0), W(u_0, u_t))$ is a walk in D containing more number of vertices than W, a contradiction to the maximality of W.

Claim 2: If there exists a (u_j, x)-walk for some j, $1 \leq j \leq t-1$, then there does not exist a (x, u_{j+1})-walk in D.

On the contrary, suppose there exist walks $W_j(u_j, x)$ and $W_{j+1}(x, u_{j+1})$. But then

$$(W(u_0, u_j), W_j(u_j, x), W_{j+1}(x, u_{j+1}), W(u_{j+1}, u_t))$$

is a (u_0, u_t)-walk containing more number of vertices than W, a contradiction to the maximality of W.

Claims 1 and 2 imply that there exist a (u_t, x)-walk, say $W'(u_t, x)$. But then $(W(u_0, u_t), W'(u_t, x))$ is a (u_0, x)-walk containing more number of vertices than W, a contradiction as before.

Therefore, $W(u_0, u_t)$ contains all the vertices of D.

(2) D contains a directed walk W containing all the vertices of $D \Rightarrow D$ is unilateral.

Let $u, v \in V(D) = V(W)$. Then W contains a directed (u, v)-subwalk if u precedes v in W or a (v, u)-subwalk if v precedes u in W. Hence D is unilateral. □

Theorem 11.4. *Let D be a simple digraph satisfying any one of the following two conditions for some integer p:*

(1) outdeg$(v) \geq p \geq 1$, for every vertex v.

(2) indeg$(v) \geq p \geq 1$, for every vertex v.

Then D contains a directed cycle of length $\geq p + 1$.

Proof. It is similar to the proof of Theorem 2.2. (Hint: choose a directed path of maximum length, say $P(x, y)$. If (1) holds, then look at $N_{out}(y)$, and if (2) holds, then look at $N_{in}(x)$.) □

Corollary. *If D is a digraph satisfying any of the conditions stated in the above theorem, then D contains a directed cycle.*

11.3 Orientation of a graph

Definition. *If G is a graph, then an **orientation** of G is a digraph $D(G)$ obtained by orienting each edge (x, y) of G from x to y or y to x but not in both directions.*

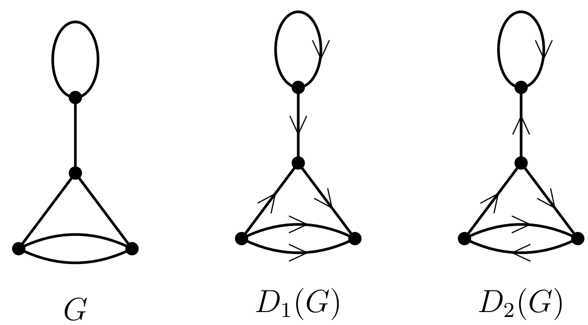


Figure 11.9: A graph G and two of its orientations.

Remark. Given a loopless graph G, there are 2^m orientations of G (some of which may be isomorphic).

Definition. An orientation $D(G)$ of a graph G is called a ***strong orientation*** if $D(G)$ is a strong digraph.

A graph G may not admit a strong orientation. For example, P_3 does not admit a strong orientation. (Why?) But K_3 admits a strong orientation.

A motivation for the definition of strong orientation is the following real-world problem.

When is it possible to make the roads of a city one-way in such a way that ***every corner of the city is reachable from every other corner ?***

Consider the road map of Figure 11.10a. It is impossible to make all the roads one-way as required. The impossibility is because of the road connecting the school and the garden.

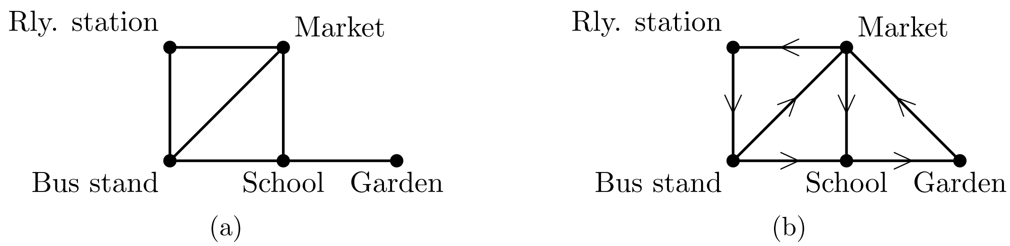


Figure 11.10: A road map.

However, if we have an additional road connecting the market and the garden, it is possible to make all the roads one-way as required (see Figure 11.10b.)

Definition. A graph G is said to be ***strongly orientable*** if it is possible to give an orientation to each edge of G so that the resulting digraph is strongly connected.

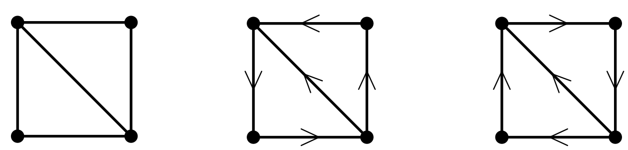


Figure 11.11: A graph G and its two strong orientations.

Thus in graph theoretic terminology, the one-way traffic problem is to characterize those graphs which are strongly orientable.

Theorem 11.5. *A connected graph G is strongly orientable if and only if G has no cut-edges.*

Proof. (1) G is strongly orientable $\Rightarrow$ G has no cut-edge.

Let D be a strong-orientation of G. Let (u, v) be an arbitrary edge of G. So, $(u, v) \in A(D)$ or $(v, u) \in A(D)$; say $(u, v) \in A(D)$. Since D is strongly connected, there exists a directed path $P(v, u)$. But then $(u, (u, v), P(v, u))$ is a cycle in G containing the edge (u, v). Hence by Theorem 2.15, (u, v) is not a cut-edge.

(2) G has no cut-edge $\Rightarrow$ G is strongly orientable.

Assume for the moment that a subgraph H with $V(G) - V(H) \neq \emptyset$ has been strongly oriented. Since $V(G) \neq V(H)$ and G is connected, there is an edge $e(u_0, u_1) \in E(G) - E(H)$, where $u_0 \in V(H)$ and $u_1 \in V(G) - V(H)$. Since e is not a cut-edge, there is a cycle $C(u_0, u_0) = (u_0, u_1, \ldots, u_p = u_0)$ in G containing e; see Figure 11.12.

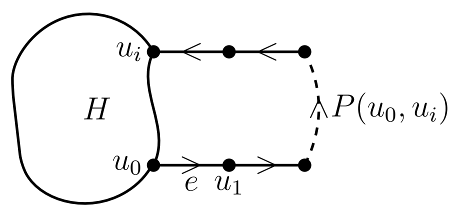


Figure 11.12: A step in the proof of Theorem 11.5.

Let u_i be the first vertex which succeeds u_1 and is in $V(H)$. (Note that u_i exists because $u_p = u_0$). Orient the path $(u_0, u_1, \ldots, u_i)$ from u to u_i so that it becomes a directed path $P(u, u_i)$. Clearly, the subdigraph H_1 which contains H and the directed path $P(u, u_i)$ is strongly connected. If we now arbitrarily orient the edges, whose end-vertices lie in $V(H) \cup \{u_0, u_1, \ldots, u_i\}$, we get a strong orientation of the subgraph induced on $V(H) \cup \{u_0, u_1, \ldots, u_i\}$. Moreover this new subdigraph contains at least one more edge. This gives us a hint to orient the whole graph G.

Since G has no cut-edges, it contains a cycle, say C. Orient C (in one of the two possible ways), so that it becomes a directed cycle C^*; obviously C^* is strongly connected. If C^* contains all the vertices of G, then we are through (after arbitrarily orienting the edges of $(E(G) - E(C))$).

If C^* does not contain all the vertices of G, extend the orientation of C^* to a larger digraph G_1 as explained above. This kind of extension can be continued until all the edges of G are oriented. □

11.4 Eulerian and Hamilton digraphs

Eulerian digraphs and Hamilton digraphs are straight forward generalizations of Eulerian graphs and Hamilton graphs. Results too are analogous. However, a few results are harder to prove and even to anticipate.

• Eulerian digraphs

Definitions.

- *A directed trail in a digraph D is called an* ***Eulerian trail*** *if it contains all the arcs in D. It can be open or closed.*
- *A digraph is called an* ***Eulerian digraph*** *if it contains a closed Eulerian trail.*

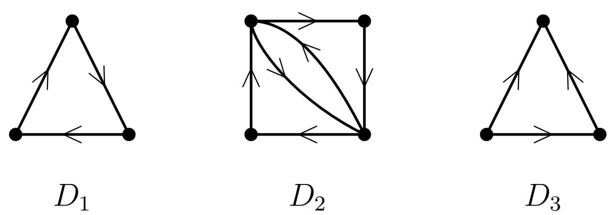


Figure 11.13: D_1 and D_2 are Eulerian digraphs and D_3 is a non-Eulerian digraph.

Theorem 11.6. *A digraph D is Eulerian if and only if*

(i) D is weakly connected, and

(ii) $outdeg(v) = indeg(v)$, for every vertex v.

Proof. It is similar to the proof of Theorem 5.1 and hence it is left as an exercise. □

• Hamilton digraphs

You may recall from Chapter 6, that there are no characterizations of Hamilton graphs. It is no surprise that there are no characterizations of Hamilton digraphs either. In this section, we prove a sufficient condition for a simple digraph to contain a directed Hamilton cycle which is analogous to Dirac condition for Hamilton graphs.

Definitions.

- *A directed path in a digraph D is called a **Hamilton directed path** if it contains all the vertices of D.*
- *A directed cycle in D is called a **Hamilton directed cycle** if contains all the vertices of D.*
- *A directed graph D is called a **Hamilton digraph** if it contains a directed Hamilton cylcle.*

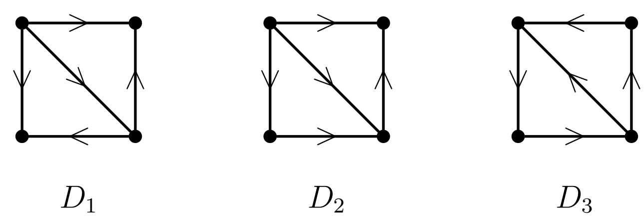


Figure 11.14: D_1 contains no directed Hamilton path. D_2 contains a directed Hamilton path but contains no directed Hamilton cycle. D_3 contains a directed Hamilton cycle and hence it is a directed Hamilton graph.

Remarks.

- D contains a directed Hamilton cycle $\Rightarrow$ D contains a directed Hamilton path.
- D contains a directed Hamilton path $\not\Rightarrow$ D contains a directed Hamilton cycle.
- D contains a directed Hamilton path $\Rightarrow$ D is unilaterally connected (Theorem 11.3.)
- D is Hamilton $\Rightarrow$ D is strong (Theorem 11.2).

Theorem 11.7 (Ghoulia-Houri, 1960). *If D is a simple digraph such that*

(i) $n \geq 3$,

(ii) $outdeg(v) \geq \frac{n}{2}$, *for every vertex* v, *and*

(iii) $indeg(v) \geq \frac{n}{2}$, *for every vertex* v,

then D contains a directed Hamilton cycle.

Proof. (Contradiction method) Assume that the result is false. Let C be a directed cycle in D containing maximum number of vertices. By our assumption, $V(D) - V(C) \neq \emptyset$. Let P be a directed path in $D - V(C)$ containing maximum number of vertices; let $P = P(a, b)$. Let $|V(C)| = k$ and $V(P) = p$. Fix the clock-wise direction to C; see Figure 11.15.

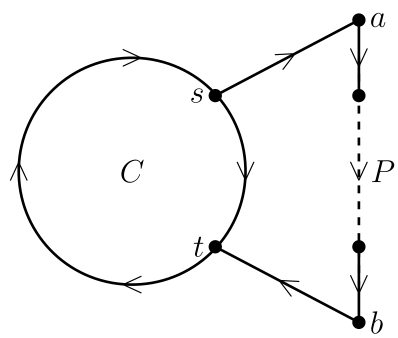


Figure 11.15: A maximum cycle C and a maximum path P in $D - V(C)$.

Clearly,

(1) $n \geq k + p$,

(2) $k > \frac{n}{2}$ (by Theorem 11.4),

(3) $N_{in}(a) \subseteq V(P - a) \cup V(C)$, by the maximality of $V(P)$.

(4) $N_{out}(b) \subseteq V(P - b) \cup V(C)$, by the maximality of $V(P)$.

Define:

$$\begin{aligned} S &= \{s \in V(C) : (s,a) \in A(D)\} \text{ and} \\ T &= \{t \in V(C) : (b,t) \in A(D)\}. \end{aligned}$$

Then

(5)

$$\begin{aligned} |S| &\geq \mathit{indeg}(a) - (p-1) \text{ (by (3))} \\ &\geq \tfrac{n}{2} - p + 1, \text{by the hypothesis} \\ &\geq \tfrac{n}{2} - (n-k) + 1 \text{ (by (1))} \\ &= -\tfrac{n}{2} + k + 1 \\ &\geq 1 \text{ (by (2))}. \end{aligned}$$

Similarly,

$$|T| \geq 1.$$

Since $|S| \geq 1$ and $|T| \geq 1$, we can choose $s \in S$ and $t \in T$ such that t is a successor of s on C and no internal vertex of the directed subpath $C[s,t]$ belongs to $S \cup T$.

Claim 1: There are at least p internal vertices in $C[s,t]$.

Else, the cycle

$$(s, a, P(a,b), b, t, C[t,s], s)$$

has more number of vertices than C, a contradiction to the maximality of $V(C)$.

Claim 2: If $(s,x) \in A(C)$, then $x \notin T$.

Else, we get a larger cycle as in the proof of Claim 1.

Therefore, there are at least $p + (|S| - 1)$ vertices on C which are not in T. Hence, using claims (1) and (2) we conclude that

$$\begin{aligned} k = |V(C)| &\geq p + (|S| - 1) + |T|, \\ &= (|S| + |T|) + p - 1, \\ &= 2(\tfrac{n}{2} - p + 1) + p - 1, \\ &\quad \text{(since } |S| \geq \tfrac{n}{2} - p + 1 \text{ and } |T| \geq \tfrac{n}{2} - p + 1\text{, see (5))} \\ &= n - p + 1. \end{aligned}$$

We have thus arrived at a contradiction to (1). □

11.5 Tournaments

Tournaments form an interesting class of digraphs. They are being independently studied. An entire book of J. W. Moon (1968) is devoted to tournaments. Further survey has been done by K. B. Reid and L. W. Bineke (1978).

Definition. *An orientation of a complete graph is called a* ***tournament****.*

So in a tournament D either $(u, v) \in A(D)$ or $(v, u) \in A(D)$ (but not both), for every pair u, v of distinct vertices. A few small tournaments are shown in Figure 11.16.

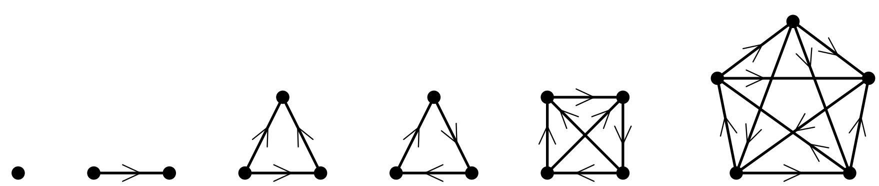


Figure 11.16: Tournaments.

Definition. *A vertex v is said to be* ***reachable*** *from a vertex u, if there is a directed path from u to v.*

Theorem 11.8. *If u is a vertex of maximum out-degree in a tournament D, then every vertex is reachable from u by a directed path of length at most 2.*

Proof. Consider the sets $N_{out}(u) = \{u_1, u_2, \ldots, u_r\}$ and $N_{in}(u) = \{v_1, v_2, \ldots, v_s\}$, where $r = outdeg(u)$ and $s = indeg(u)$. Since D is a tournament, $V(D) - \{u\} = N_{out}(u) \cup N_{in}(u)$ and $N_{out}(u) \cap N_{in}(v) = \emptyset$.

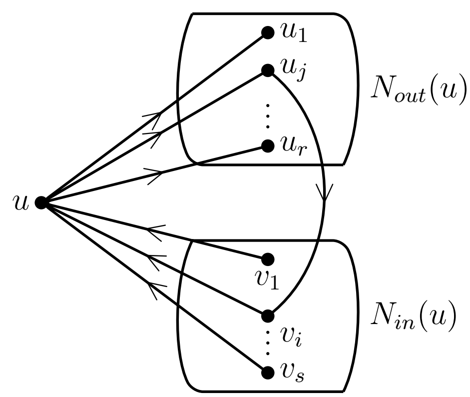


Figure 11.17

Clearly, every u_j is reachable from u by a path of length 1. We next assert that for every v_i $(1 \leq i \leq s)$, there is some u_j $(1 \leq j \leq r)$ such that (u_j, v_i) is an arc in D; so that v_i is reachable from u by a path of length 2, namely (u, u_j, v_i). Assume that our assertion is false for some v_i. Then (v_i, u_j) is an arc for every j $(1 \leq j \leq r)$. Hence,

$$N_{out}(v_i) \supseteq \{u_1, u_2, \ldots, u_r\} \cup \{u\}.$$

Thus, $outdeg(v_i) \geq outdeg(u) + 1$, a contradiction to the maximality of $outdeg(u)$. Hence, our assertion indeed holds and the theorem follows. $\square$

Theorem 11.9. *Every tournament D contains a directed Hamilton path.*

Proof. We prove the theorem by induction on n. If $n = 1$ or 2, then the theorem is obvious. Assume that a tournament contains a directed Hamilton path if it has $n-1$ vertices and let D contain n vertices.

Let $v \in V(D)$ and consider the tournament $D - v$. By induction hypothesis, $D - v$ contains a directed Hamilton path say $(v_1, v_2, \ldots, v_{n-1})$; see Figure 11.18.

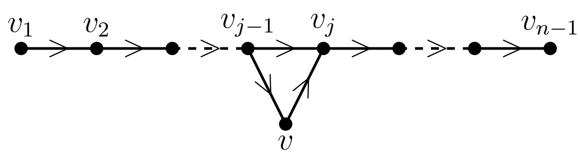


Figure 11.18: Extension of a path.

We make 3 cases and prove that D contains a directed Hamilton path in each case.

Case 1: (v, v_1) is an arc D.

Clearly, $(v, v_1, v_2, \ldots, v_{n-1})$ is a directed Hamilton path in D.

Case 2: (v, v_1) is not an arc (and hence (v_1, v) is an arc) but there is some i $(2 \leq i \leq n-1)$ such that (v, v_i) is an arc.

Let j $(2 \leq j \leq n-1)$ be the smallest integer such that (v, v_j) is an arc in D. This means that (v, v_{j-1}) is not an arc; and hence (v_{j-1}, v) is an arc. But then

$$(v_1, v_2, \ldots, v_{j-1}, v, v_j, v_{j+1}, \ldots, v_{n-1})$$

is a directed Hamilton path in D.

Case 3: There is no i $(1 \leq i \leq n-1)$, such that (v, v_i) is an arc.

This means, in particular, (v_{n-1}, v) is an arc in D. But then

$$(v_1, v_2, \ldots, v_{n-1}, v)$$

is a directed Hamilton path in D. □

Corollary. *Every tournament is unilaterally connected.*

Proof. A consequence of Theorem 11.3. □

A tournament need not contain a directed Hamilton cycle. In Figure 11.16, the second and third tournaments are non-Hamilton whereas the others are Hamilton. However, every strong tournament on at least three vertices, contains a directed Hamilton cycle. We shall prove a stronger assertion.

Theorem 11.10. *Every strong tournament D on n (≥ 3) vertices contains a directed cycle of length k, for every k, $3 \leq k \leq n$.*

Proof. We first prove that D contains a directed 3-cycle and next show that, if D contains a directed k-cycle, for some k $(3 \leq k \leq n-1)$, then it contains a directed $(k+1)$-cycle; so that D contains a directed k-cycle, for every k, $3 \leq k \leq n$.

(1) D contains a directed 3-cycle.

Let $v \in V(D)$ and consider $N_{out}(v)$ and $N_{in}(v)$; see Figure 11.19.

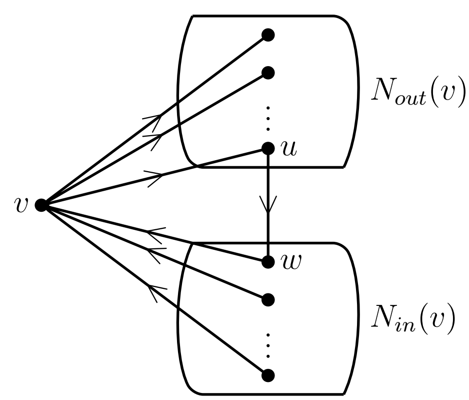


Figure 11.19: A 3-cycle in a strong tournament.

Since D is strongly connected, $N_{out}(v)$ and $N_{in}(v)$ are non-empty. Moreover, there is an arc (u, w) in D where $u \in N_{out}(v)$ and $w \in N_{in}(v)$ because every directed path from v to a vertex $w \in N_{in}(v)$ must pass through a vertex in $N_{out}(v)$. But then (v, u, w, v) is a directed cycles of length 3.

(2) Let $C(v_1, v_2, \ldots, v_k, v_1)$ be a directed cycle of length k in D, for some k $(3 \leq k \leq n-1)$.

We make two cases and in each case construct a directed $(k+1)$-cycle.

Case 1: There is a vertex $u \in D - V(C)$ such that (v_i, u) and (u, v_j) are arcs in D for some i and j; without loss of generality, let $(v_1, u) \in A(D)$.

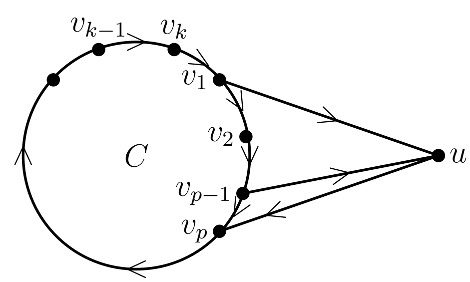


Figure 11.20: An extension of a k-cycle to a $(k+1)$-cycle.

Let p $(2 \leq p \leq k)$ be the smallest integer such that $(u, v_p) \in A(D)$. So, $(v_{p-1}, u) \in A$. Then

$$(v_1, v_2, \ldots, v_{p-1}, u, v_p, v_{p+1}, \ldots, v_k, v_1)$$

is directed cycle of length $(k+1)$ in D; see Figure 11.20.

If the assumption made in Case 1 does not hold, then the following must hold.

Case 2: If $u \in D - V(C)$, then one of the following holds.

(a) $(u, v_i) \in A(D)$, for every i, $1 \leq i \leq k$,

(b) $(v_i, u) \in A(D)$, for every i, $1 \leq i \leq k$.

Define

$X = \{u \in D - V(C); (v_i, u) \in A(D)$, for every $i, 1 \leq i \leq k\}$, and

$Y = \{w \in D - V(C); (w, v_i) \in A(D)$, for every $i, 1 \leq i \leq k\}$.

Since D is strong and since for every $u \in D - V(C)$ either (a) or (b) holds, it follows that $X \neq \emptyset$, $Y \neq \emptyset$, $X \cap Y = \phi$ and $X \cup Y = V(D) - V(C)$.

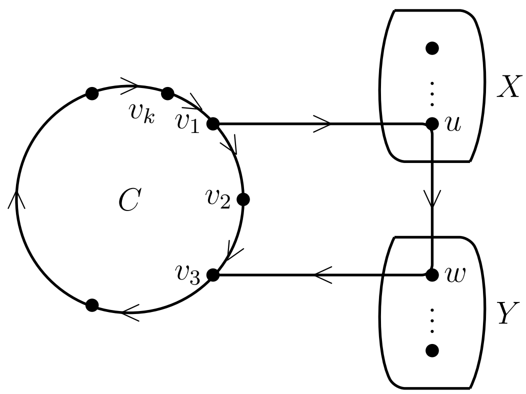


Figure 11.21

There are vertices $u \in X$ and $w \in Y$ such that (u, w) is an arc in D, because every directed path from a vertex in C to a vertex in Y must pass through a vertex in X. Now, since $u \in X$ and $w \in Y$, (v_1, u) and (w, v_3) are arcs in D. But then $(v_1, u, w, v_3, v_4, \ldots, v_k, v_1)$ a directed cycle of length $k + 1$ in D. □

Corollary. *A tournament is strong if and only if it contains a directed Hamilton cycle.*

Proof. A consequence of Theorems 11.2 and 11.10. □

Exercises

1. Draw all the non-isomorphic strong simple digraphs on 4 vertices and 5 arcs.

2. If D is a digraph in which every vertex has out-degree 1, then show that D has exactly one directed cycle.

3. Draw (as many as you can) simple non-isomorphic digraphs on 7 vertices in which every vertex has out-degree 1 and every vertex has in-degree 1. (Do you recognize any relation on such digraphs and partition of 7?).

4. Let G be a k-regular graph on vertices $v_1, v_2, \ldots, v_n$ and let D be an orientation of G. Show that $\sum_{i=1}^{n}(outdeg(v_i))^2 = \sum_{i=1}^{n}(indeg(v_i))^2$.

5. A simple digraph D is called k***-regular*** if $outdeg(v) = indeg(v) = k$, for every vertex $v \in V(D)$.

   (a) Draw a 2-regular digraph on 5 vertices.

   (b) Let k and n be integers such that $0 \leq k < n$. Describe a construction to obtain a k-regular digraph on n vertices.

6. Prove or disprove: For every $n \geq 1$, there is a simple digraph on n vertices in which every vertex has odd out-degree.

7. (a) Construct pairs (D_1, D_2) of simple digraphs on n vertices for $n = 2, 3, 4$ where $V(D_1) = \{u_1, u_2, \ldots, u_n\}$ and $V(D_2) = \{v_1, v_2, \ldots, v_n\}$ such that

      i. $D_1 \not\simeq D_2$.

      ii. $D_1 - u_i \simeq D_2 - v_i$, $i = 1, 2, \ldots, n$.

   (b) Verify that the digraphs D_1 and D_2 shown below have the properties (i) and (ii) stated above.

8. If D is a weakly connected digraph, then show that $m \geq n - 1$.

9. If D is a digraph with $m \geq (n-1)(n-2) + 1$, then show that D is weakly connected.

10. If D is a strongly connected simple digraph, then show that $n \leq m \leq n(n-1)$.

11. If D is a simple digraph with $m \geq (n-1)^2 + 1$, then show that D is strongly connected.

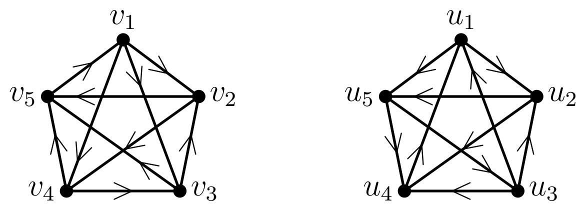


Figure 11.22

12. Give an example of a simple digraph with the following properties:

    (a) $m = (n-1)(n-2)$, which is disconnected;

    (b) $m = n$, which is strongly connected;

    (c) $m = n - 1$, which is weakly connected;

    (d) $m = n - 1$, which is unilaterally connected;

    (e) $m = (n-1)^2$, which is not strongly connected.

    In view of the existence of these examples, discuss the merits of the bounds stated in Exercises 8 to 11.

13. Show that a digraph is strong if and only if its converse digraph is strong. (The ***converse digraph*** $\overleftarrow{D}$ of D has vertex set $V(\overleftarrow{D}) = V(D)$. $(u, v) \in A(\overleftarrow{D})$ iff $(v, u) \in A(D)$.)

14. Use the proof of Theorem 11.5 to give a strong orientation to the following graphs.

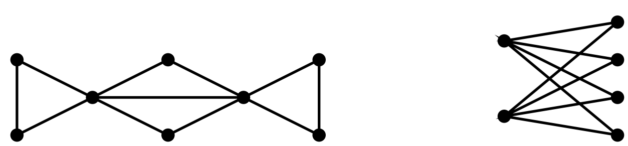


Figure 11.23

15. If D is a weakly connected simple digraph such that $D - v$ is strong for some vertex $v \in V(D)$, then show that D is unilaterally connected.

16. The adjacency matrix $M = [a_{ij}]$ of a digraph D on n vertices $v_1, v_2, \ldots, v_n$ is a $n \times n$ matrix where $a_{ij} = 1$ if (v_i, v_j) is an arc in D and $a_{ij} = 0$, otherwise. Show that

    (a) $\sum_{j=1}^{n} a_{ij} = outdeg(v_i)$, $t = 1, 2, \ldots, n$;

    (b) $\sum_{j=1}^{n} a_{ij} = indeg(v_i)$, $t = 1, 2, \ldots, n$;

    (c) The (i, j)-th entry $[M^p]_{ij}$ in M^p is the number of directed walks of length p from v_i to v_j.

17. Every graph G has an orientation D such that $|outdeg(v) - indeg(v)| \leq 1$, for every vertex v.

18. Let D be a digraph and let $r = \max\limits_{v \in V}\{outdeg(v), indeg(v)\}$. Prove that there is an r-regular digraph H such that D is a subdigraph of H.

19. Prove that there exist regular tournaments of every odd order but there are no regular tournaments of even order.

20. Show that in a tournament there is at most one vertex of out-degree zero and at most one vertex of in-degree zero.

21. Let D be a strong tournament. Given any k, $1 \leq k \leq n - 3$, show that there exists a set $S \subseteq V(D)$ of k vertices such that $D - S$ is strongly connected.

22. (a) Draw a tournament on 5 vertices in which every vertex has the same out-degree.

    (b) If a tournament has n vertices and every vertex has out-degree d, find d.

23. Draw:

    (a) A unilaterally connected (but not strongly connected) tournament on 5 vertices.

    (b) A strongly connected tournament on 5 vertices.
    Justify that your tournaments indeed have the required properties.

24. Show that a tournament D is not strongly connected if and only if there is a partition of $V(D)$ into two subsets A and B such that every arc in between A and B is of the form (u, v), where $u \in A$ and $v \in B$.

25. A tournament D is called a ***transitive tournament*** if $(u, w) \in A(D)$ whenever (u, v) and $(v, w) \in A(D)$. Show that a tournament is transitive if and only if the vertices of D can be ordered $v_1, v_2, \ldots, v_n$ such that $outdeg(v_1) = 0$, $outdeg(v_2) = 1, \ldots, outdeg(v_n) = n - 1$.

26. Show that a tournament is transitive if and only if it does not contain any directed cycle.

27. If T is a tournament on n vertices $v_1, v_2, \ldots, v_n$ with $outdeg(v_i) = s_i,\ i = 1, 2, \ldots, n$ then show that

    (a) $\sum_{i=1}^{n} s_i \geq \dfrac{k(k-1)}{2}\ 1 \leq k < n.$

    (b) $\sum_{i=1}^{n} s_i = \dfrac{n(n-1)}{2}$;

    (c) $\sum_{i=1}^{n} s_i^2 = \sum_{i=1}^{n} (n - 1 - s_i)^2$;

28. A vertex v in a tournament T is called a winner if every vertex can be reached from v by a directed path of length ≤ 2. Show the following:

    (a) No tournament has exactly two winners.

    (b) For every $n \neq 2, 4$, there is a tournament of order n in which every vertex is a winner.

List of Books

- **Old Classics**

[1] C. Berge. *Graphs and Hypergraphs.* North Holland, 1973.

[2] F. Harary. *Graph Theory.* Addison-Wesley, Reading MA, 1969.

[3] D. König. *Theorie der endlichen und unendlichen Graphen.* Akademische Verlagsgesellschaft, 1936.

[4] O. Ore. *Theory of Graphs*, volume XXXVIII of *American Mathematical Society Colloqium Publications.* American Mathematical Society, 1962.

- **Text Books**

[5] J. A. Bondy and U. S. R. Murty. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics.* Springer, 1st edition, 2008.

[6] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications.* http://www.math.jussieu.fr/~jabondy/books/gtwa/gtwa.html, 2008.

[7] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics.* Springer, 2nd edition, 2000.

[8] L. Lovász. *Combinatorial Problems and Exercises.* Akademiai Kiado and North Holland, 1979.

[9] D.B. West. *Introduction to Graph Theory.* Prentice Hall of India, 2005.

- **Books on Selected Topics**

[10] J. Akiyama and M. Kano. *Factors and Factorizations of Graphs: Proof Techniques in Factor Theory*, volume 2031 of *Lecture Notes in Mathematics*. Springer, 2010.

[11] R.B. Bapat. *Graphs and Matrices*. Hindustan Book Agency, 2010.

[12] L.W. Beineke and R.J. Wilson. *Selected Topics in Graph Theory Vols. 1,2 and 3*. Academic Press, 1978(1988).

[13] N. Biggs. *Algebraic Graph Theory*, volume 67 of *Cambridge Tracts in Mathematics*. Cambridge University Press, 1974.

[14] D. Bonchev and D.H. Rouvray. *Chemical Graph Theory: Introduction and Fundamentals*. Abacus Press/Gordon and Breach Science Publishers, 1991.

[15] A. Brandstädt, V.B. Le, and J.P. Spinrad. *Graph Classes - A Survey*, volume 3. SIAM, Monographs on Discrete Mathematics and Applications, Philadelphia, 1999.

[16] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 3rd edition, 2009.

[17] J. Duato, S. Yalamanchili, and L.-M. Ni. *Interconnection Networks: An Engineering Approach*. Morgan Kauffman, 2003.

[18] H. Fleischner. *Eulerian Graphs and Related Topics, Part 1*, volume 45 of *Ann. Discrete Mathemaics*. North-Holland, Amsterdam, 1990.

[19] H. Fleischner. *Eulerian Graphs and Related Topics, Part 2*, volume 50 of *Ann. Discrete Mathemaics*. North-Holland, Amsterdam, 1991.

[20] C. Godsil, and G. Royle. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer, 2001.

[21] R.L. Graham, B.L. Rothschild, and J.H. Spencer. *Ramsey Theory*. Wiley Interscience in Dicrete Mathematics, 1980.

[22] T. R. Jensen and B. Toft. *Graph Coloring Problems*. Wiley-Interscience, New York, 1995.

[23] L. Lovász and D. Plummer. *Matching Theory (Ann. of Disc. Math,* **29***)*. Akademiai Kiado and North Holland, 1986.

[24] J.W. Moon. *Topics on Tournaments*. Holt, Rinehart and Winston, 1968.

[25] N. Trinajstic. *Chemical Graph Theory*. Mathematical Chemistry Series. CRC Press, 1992.