GRAPH THEORY

Dr Sudev Naduvath

Associate Professor
Department of Mathematics
CHRIST (Deemed to be University)
Bengaluru-560029, Karnataka, India.

Graph Theory
Author : Dr Sudev Naduvath

Copyright © 2020 The Author

Mathematics Subject Classification 2010 : 05CXX

Published by

Centre for Studies in Discrete Mathematics
Thrissur, Kerala, India.
Email: csdmthrissur@gmail.com,
`http://www.csdm.org.in`

First printing, March 2020

To my beloved Parents

Contents

****

Chapter 1

Introduction to Graphs

****

Graph Theory is a well-known area of discrete mathematics which deals with the study of graphs. A graph may be considered as a mathematical structure that is used for modelling the pairwise relations between objects.

Graph Theory has many theoretical developments and applications not only to different branches of mathematics, but also to various other fields of basic sciences, technology, social sciences, computer science etc. Graphs are widely used as efficient tools to model many types of practical and real-world problems in physical, biological, social and information systems. Graph-theoretical models and methods are based on mathematical combinatorics and related fields.

1.1 Basic Definitions

Definition 1.1.1 (Graph)**.** A *graph* G can be considered as an ordered triple (V, E, ψ), where

(i) $V = \{v_1, v_2, v_3, \ldots\}$ is called the *vertex set* of G and the elements of V are called the *vertices* (or *points* or *nodes*);

(ii) $E = \{e_1, e_2, e_3, \ldots\}$ is the called the *edge set* of G and the elements of E are called *edges* (or *lines* or *arcs*); and

(iii) ψ is called the *adjacency relation*, defined by $\psi : E \to V \times V$, which defines the association between each edge with the vertex pairs of G.

Usually, the graph is denoted as $G = (V, E)$. The vertex set and edge set of a graph G are also written as $V(G)$ and $E(G)$ respectively.

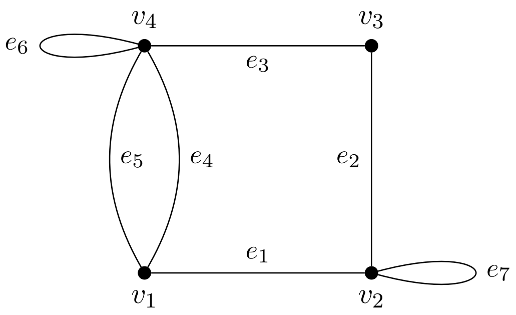


Figure 1.1: An example of a graph

If two vertices u and v are the (two) end points of an edge e, then we represent this edge by uv or vu. If $e = uv$ is an edge of a graph G, then we say that u and v are *adjacent vertices* in G and that e *joins* u and v. In such cases, we also say that u and v are adjacent to each other.

Given an edge $e = uv$, the vertex u and the edge e are said to be *incident with* each other and so are v and e. Two edges e_i and e_j are said to be *adjacent edges* if they are incident with a common vertex.

Definition 1.1.2 (Order and Size of a Graph)**.** The *order* of a graph G, denoted by $\nu(G)$, is the number of its vertices and the *size* of G, denoted by $\epsilon(G)$, is the number of its edges.

A graph with p-vertices and q-edges is called a (p,q)*-graph.* The $(1,0)$-graph is called a *trivial graph.* That is, a trivial graph is a graph with a single vertex. A graph without edges is called an *empty graph* or a *null graph.* The following figure illustrates a null graph of order 5.

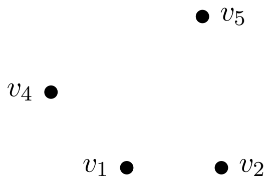


Figure 1.2: Null graph of order 5.

Definition 1.1.3 (Finite and Infinite Graphs)**.** A graph with a finite number of vertices as well as a finite number of edges is called a *finite graph.* Otherwise, it is an *infinite graph.*

Definition 1.1.4 (Self-loop)**.** An edge of a graph that joins a node to itself is called *loop* or a *self-loop.* That is, a loop is an edge uv, where $u = v$.

Definition 1.1.5 (Parallel Edges)**.** The edges connecting the same pair of vertices are called *multiple edges* or *parallel edges.*

In Figure 1.2, the edges e_6 and e_7 are loops and the edges e_4 and e_5 are parallel edges.

Definition 1.1.6 (Simple Graphs and Multigraphs)**.** A graph G which does not have loops or parallel edges is called a *simple graph.* A graph which is not simple is generally called a *multigraph.*

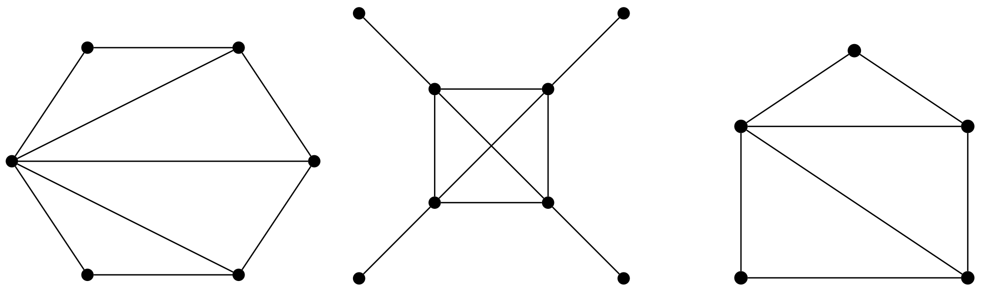


Figure 1.3: Some examples of simple graphs

1.2 Degrees and Degree Sequences in Graphs

Definition 1.2.1 (Degree of a vertex)**.** The number of edges incident on a vertex v, with self-loops counted twice, is called the *degree* of the vertex v and is denoted by $deg_G(v)$ or $deg(v)$ or simply $d(v)$.

Definition 1.2.2 (Isolated vertex)**.** A vertex having no incident edge is called an *isolated vertex.* In other words, isolated vertices are those with zero degree.

Definition 1.2.3 (Pendant vertex)**.** A vertex of degree 1, is called a *pendent vertex* or an end vertex.

Definition 1.2.4 (Internal vertex)**.** A vertex, which is neither a pendent vertex nor an isolated vertex, is called an *internal vertex* or an *intermediate vertex.*

Definition 1.2.5 (Minimum and Maximum Degree of a Graph)**.** The *maximum degree of a graph* G, denoted by $\Delta(G)$, is defined to be $\Delta(G) = \max\{d(v) : v \in V(G)\}$. Similarly, the *minimum degree of a graph* G, denoted by $\delta(G)$, is defined to be $\delta(G) = \min\{d(v) : v \in V(G)\}$. Note that for any vertex v in G, we have $\delta(G) \leq d(v) \leq \Delta(G)$.

The following theorem is a relation between the sum of degrees of vertices in a graph G and the size of G.

Theorem 1.2.6. *In a graph G, the sum of the degrees of the vertices is equal to twice the number of edges. That is, $\sum\limits_{v\in V(G)} d(v) = 2\epsilon$.*

Proof. Let $S = \sum\limits_{v\in V(G)} d(v)$. Notice that in counting S, we count each edge exactly twice. That is, every edge contributes degree 1 each to both of its end vertices and a loop provides degree 2 to the vertex it incidents with. Hence 2 to the sum of degrees of vertices in G. Thus, $S = 2|E| = 2\epsilon$. □

The above theorem is usually called the *first theorem of graph theory.* It is also known as the *hand shaking lemma.* The following two theorems are immediate consequences of the above theorem.

Theorem 1.2.7. *For any graph G, $\delta(G) \leq \frac{2|E|}{|V|} \leq \Delta(G)$.*

Proof. By Theorem-1, we have $2\epsilon = \sum\limits_{v\in V(G)} d(v)$. Therefore, note that $\frac{2|E|}{|V|} = \frac{\sum d(v)}{|V|}$, the average degree of G. Therefore, $\delta(G) \leq \frac{2|E|}{|V|} \leq \Delta(G)$. □

Theorem 1.2.8. *For any graph G, the number of odd degree vertices is always even.*

Proof. Let $S = \sum\limits_{v\in V(G)} d(v)$. By Theorem 1.2.6, we have $S = 2\epsilon$ and hence S is always even. Let V_1 be the set of all odd degree vertices and V_2 be the set of all even degree vertices in G. Now, let $S_1 = \sum\limits_{v\in V_1} d(v)$ and $S_2 = \sum\limits_{v\in V_2} d(v)$. Note that S_2, being the sum of even integers, is also an even integer.

We also note that $S = S_1 + S_2$ (since V_1 and V_2 are disjoint sets and $V_1 \cup V_2 = V$). Therefore, $S_1 = S - S_2$. Being the difference between two even integers, S_1 is also an even integer. Since V_1 is a set of odd degree vertices, S_1 is even only when the number of elements in V_1 is even. That is, the number of odd degree vertices in G is even, completing the proof. □

Definition 1.2.9 (Degree Sequence)**.** The *degree sequence* of a graph of order n is the n-term sequence (usually written in descending order) of the vertex degrees. In Figure-1, $\delta(G) = 2$, $\Delta(G) = 5$ and the degree sequence of G is $(5,4,3,2)$.

Definition 1.2.10 (Graphical Sequence)**.** An integer sequence is said to be *graphical* if it is the degree sequence of some simple graphs. A simple graph G is said to be the *graphical realisation* of an integer sequence S if S the degree sequence of G.

Problem 1.2.11. Is it possible to draw a graph with four vertices v_1, v_2, v_3, v_4 with degrees $3,5,2,1$ respectively? If not, why?

Solution: By Theorem 1.2.6, the sum of vertex degrees in any graph is even. But here, the degree sum is $3+5+2+1 = 11$. Therefore, it is not possible to draw such a graph. □

Problem 1.2.12. Is the sequence $S = \langle 5,4,3,3,2,2,2,1,1,1,1\rangle$ graphical? Justify your answer.

Solution: The sequence $S = \langle a_i \rangle$ is graphical if every element of S is the degree of some vertex in a graph. For any graph, we know that $\sum_{v \in V(G)} d(v) = 2|E|$, an even integer. Here, $\sum a_i = 25$, not an even number. Therefore, the given sequence is not graphical. □

Problem 1.2.13. Is the sequence $S = \langle 9,9,8,7,7,6,6,5,5\rangle$ graphical? Justify your answer.

Solution: The sequence $S = \langle a_i \rangle$ is graphical if every element of S is the degree of some vertex in a graph. For any graph, we know that $\sum_{v \in V(G)} d(v) = 2|E|$, an even integer. Here, $\sum a_i = 62$, an even number. But note that the maximum degree that a vertex can attain in a graph of order n is $n-1$. If S were graphical, the corresponding graph would have been a graph on 9 vertices and have $\Delta(G) = 9$. Therefore, the given sequence is not graphical. □

Problem 1.2.14. Is the sequence $S = \langle 9,8,7,6,6,5,5,4,3,3,2,2\rangle$ graphical? Justify your answer.

Solution: The sequence $S = \langle a_i \rangle$ is graphical if every element of S is the degree of some vertex in a graph. For any graph, we know that $\sum_{v \in V(G)} d(v) = 2|E|$, an even integer. Here, we have $\sum a_i = 60$, an even number. Also, note that the all elements in the sequence are less than the number of elements in that sequence. Therefore, the given sequence is graphical and the corresponding graph is drawn below. □

Problem 1.2.15. Show that in any group of two or more people, there are always at least two with exactly the same number of friends inside the group.

Solution: Consider each member of the group as a vertex and join the two vertices if the corresponding members in the group are friends and complete the construction

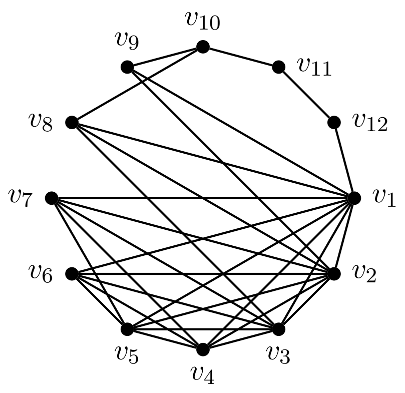


Figure 1.4: Graphical realisation of the degree sequence S.

of the graph G accordingly. Then, the number of friends a person has in the group is the degree of the corresponding vertex. Hence, we need only to prove that at least two points of G have the same degree. For this, let $V(G) = \{v_1, v_2, v_3 \ldots, v_n\}$, where n is the size of the group. Clearly, $0 \leq d(v_i) \leq n-1$ for $v_i \in V(G)$. The degree sequence of G consists of n elements with highest possible entry is $n-1$. Hence, at least two vertices of G must have the same degree. That is, at least two members in the group has same number of friends. □

1.2.1 Neighbourhoods

Definition 1.2.16 (Neighbourhood of a Vertex)**.** The *neighbourhood* (or *open neighbourhood*) of a vertex v, denoted by $N(v)$, is the set of vertices adjacent to v. That is, $N(v) = \{x \in V : vx \in E\}$. The *closed neighbourhood* of a vertex v, denoted by $N[v]$, is simply the set $N(v) \cup \{v\}$.

Then, for any vertex v in a graph G, we have $d_G(v) = |N(v)|$. A special case is a loop that connects a vertex to itself; if such an edge exists, the vertex is said to belong to its own neighbourhood.

Given a set S of vertices, we define the neighbourhood of S, denoted by $N(S)$, to be the union of the neighbourhoods of the vertices in S. Similarly, the closed neighbourhood of S, denoted by $N[S]$, is defined to be $S \cup N(S)$.

Neighbourhoods are widely used to represent graphs in computer algorithms, in terms of the adjacency list and adjacency matrix representations. Neighbourhoods are also used in the clustering coefficient of graphs, which is a measure of the average density of its neighbourhoods. In addition, many important classes of graphs may be defined by properties of their neighbourhoods, or by symmetries that relate neighbourhoods to each other.

1.3 Subgraphs and Spanning Subgraphs

Definition 1.3.1 (Subgraph of a Graph)**.** A graph $H(V_1, E_1)$ is said to be a *subgraph* of a graph $G(V, E)$ if $V_1 \subseteq V$ and $E_1 \subseteq E$.

Definition 1.3.2 (Spanning Subgraph of a Graph)**.** A graph $H(V_1, E_1)$ is said to be a *spanning subgraph* of a graph $G(V, E)$ if $V_1 = V$ and $E_1 \subseteq E$.

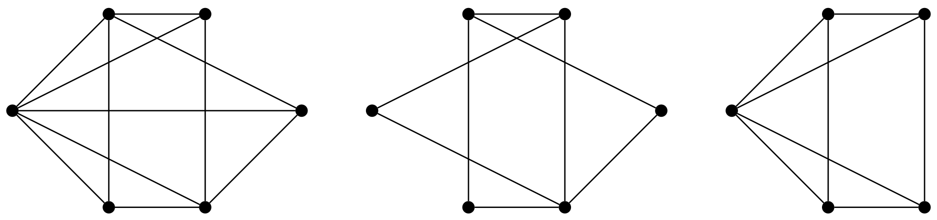


Figure 1.5: Examples of Subgraphs

In the above figure, the second graph is a spanning subgraph of the first graph, while the third graph is a subgraph of the first graph.

1.3.1 Induced Subgraphs

Definition 1.3.3 (Induced Subgraph)**.** Suppose that V' be a subset of the vertex set V of a graph G. Then, the subgraph of G whose vertex set is V' and whose edge set is the set of edges of G that have both end vertices in V' is denoted by $G[V]$ or $\langle V \rangle$ called an *induced subgraph* of G.

Definition 1.3.4 (Edge-Induced Subgraph)**.** Suppose that E' be a subset of the edge set V of a graph G. Then, the subgraph of G whose edge set is E' and whose vertex set is the set of end vertices of the edges in E' is denoted by $G[E]$ or $\langle E \rangle$ called an *edge-induced subgraph* of G.

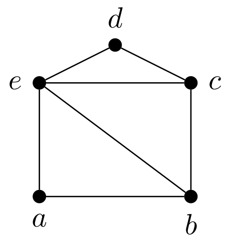
Figure 1.6 depicts an induced subgraph and an edge induced subgraph of a given graph.

1.4 Fundamental Graph Classes

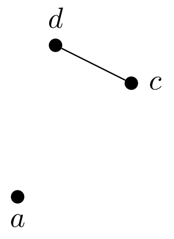
1.4.1 Complete Graphs

Definition 1.4.1 (Complete Graphs)**.** A *complete graph* is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge. A complete graph on n vertices is denoted by K_n.

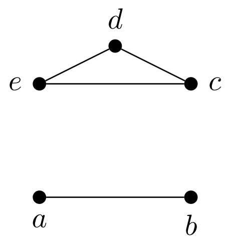
Problem 1.4.2. Show that a complete graph K_n has $\frac{n(n-1)}{2}$ edges.



(a) The graph G



(b) The induced subgraph $G[a,c,d]$



(c) The induced subgraph $G[ab,cd,ce,de]$

Figure 1.6: Induced and edge-induced subgraphs of a graph G.

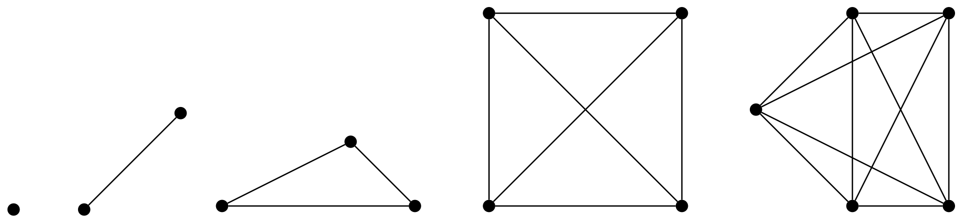


Figure 1.7: First few complete graphs

Solution: Note that any two vertices in a complete graph are adjacent to each other. Hence, the number of edges in a complete graph is equal to the number of distinct pairs of vertices in it. Therefore, the number of such pairs of vertices in K_n is $\binom{n}{2} = \frac{n(n-1)}{2}$. That is, the number of edges in K_n is $\frac{n(n-1)}{2}$. □

We can write an alternate solution to this problem as follows:

Solution: Note that every vertex in a complete graph K_n is adjacent to all other $n-1$ vertices in K_n. That is, $d(v) = n-1$ for all vertices in K_n. Since K_n has n vertices, we have $\sum \lim\limits_{v \in V(K_n)} d(v) = n(n-1)$. Therefore, by the first theorem on graph theory, we have $2|E(K_n)| = n(n-1)$. That is, the number of edges in K_n is $\frac{n(n-1)}{2}$. □

Problem 1.4.3. Show that the size of every graph of order n is at most $\frac{n(n-1)}{2}$.

Solution: Note that every graph on n vertices is a spanning subgraph of the complete graph K_n. Therefore, $E(G) \subseteq E(K_n)$. That is, $|E(G)| \leq |E(K_n)| = \frac{n(n-1)}{2}$. That is, any graph of order n can have at most $\frac{n(n-1)}{2}$ edges. □

Definition 1.4.4 (Independent Vertices)**.** Two vertices are of a graph G are said to be *independent* in G if they are not adjacent to each other.

Definition 1.4.5 (Independent Set)**.** A set $S \subseteq V(G)$ of a graph G is said to be an *independent set* of G if no two vertices in S are adjacent to each other.

Definition 1.4.6 (Independence Number)**.** The *independence number* of a graph G, denoted by $\alpha(G)$, is said the number of vertices in a maximal *independent set* of G.

1.4.2 Bipartite Graphs

Definition 1.4.7 (Bipartite Graphs)**.** A graph G is said to be a *bipartite graph* if its vertex set V can be partitioned into two sets, say V_1 and V_2, such that no two vertices in the same partition can be adjacent. Here, the pair (V_1, V_2) is called the *bipartition* of G.

In other words, a bipartite graph is a graph whose vertex set can be partitioned into two independent sets.

Figure 1.8 gives some examples of bipartite graphs. In all these graphs, the white vertices belong to the same partition, say V_1 and the black vertices belong to the other partition, say V_2.

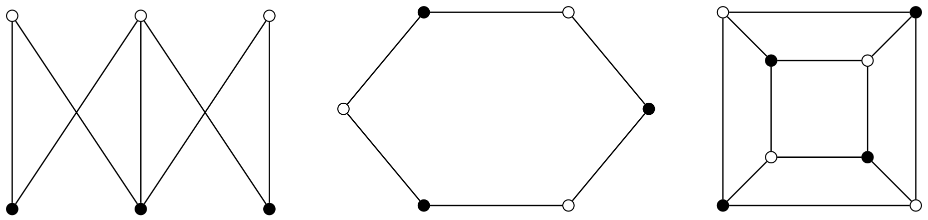


Figure 1.8: Examples of bipartite graphs

Note that for any bipartite graph G with bipartition (X,Y), we have

$$|E(G) = \sum_{v\in X} d(v) = \sum_{v\in Y} d(v).$$

Definition 1.4.8 (Complete Bipartite Graphs)**.** A bipartite graph G is said to be a *complete bipartite graph* if every vertex of one partition is adjacent to every vertex of the other. A complete bipartite graph with bipartition (X,Y) is denoted by $K_{|X|,|Y|}$ or $K_{a,b}$, where $a = |X|, b = |Y|$.

The following graphs are also some examples of complete bipartite graphs. In these examples also, the vertices in the same partition have the same colour.

Problem 1.4.9. Show that a complete bipartite graph $K_{a,b}$ has ab edges.

Solution: Let $K_{a,b}$ be a complete bipartite graph with bipartition (X,Y). Note that all a vertices in X have the same degree b and all b vertices in Y have the same degree a. Therefore, $\sum\limits_{v\in V(K_{a,b})} d(v) = ab + ba = 2ab$. By the first theorem on graph theory, we have $2|E(K_{a,b})| = 2ab$. That is, $|E(K_{a,b})| = ab$. □

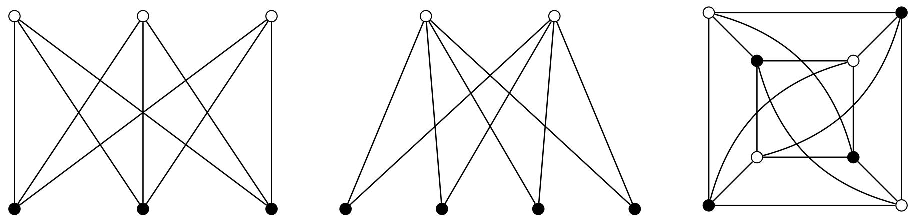


Figure 1.9: Examples of complete bipartite graphs

Theorem 1.4.10. *The complete graph K_n can be expressed as the union of k bipartite graphs if and only if $n \leq 2^k$.*

Proof. First assume that K_n can be expressed as the union of k bipartite graphs. We use the method of induction on k. First let $k = 1$. Note that K_n contains triangle K_3 (and K_3 is not bipartite) except for $n \leq 2$. Therefore, the result is true for $k = 1$.

Now assume that $k > 1$ and the result holds for all complete graphs having fewer than k complete bipartite components. Now assume that $K_n = G_1 \cup G_2 \cup \ldots, \cup G_k$, where each G_i is bipartite. Partition the vertex set V into two components such that the graph G_k has no edge within X or within Y. The union of other $k-1$ bipartite subgraphs must cover the complete subgraphs induced by X and Y. Then, by Induction hypothesis, we have $|X| \leq 2^{k-1}$ and $|YX| \leq 2^{k-1}$. Therefore, $n = |X| + |Y| \leq 2^{k-1} + 2^{k-1} = 2^k$. Therefore, the necessary part follows by induction. $\square$

Problem 1.4.11. A graph has the property that every edge of it joins an odd degree vertex with an even vertex. Then, show that

(i) G has even number of edges.

(ii) G is bipartite.

Solution: Here, we can partition the vertex set of G into two sets as follows:

$$\begin{aligned} X &= \{v \in V(G) : d_G(v) \text{is even}\} \\ Y &= \{v \in V(G) : d_G(v) \text{is odd}\}. \end{aligned}$$

Since every edge of G joins an odd degree vertex with an even vertex, no two vertices in X and no two vertices in Y are adjacent. Therefore, G bipartite.

Also, $|E(G)| = \sum\limits_{v \in X} d(v) = \sum\limits_{v \in Y} d(v)$, which is an even number.

1.4.3 Multipartite Graphs

Definition 1.4.12 (k-Partite Graph)**.** A *k-partite graph* is a graph whose vertex set can be partitioned into k sets, such that no two vertices in the same partition are adjacent.

In other words, a k-partite graph is a graph whose vertex set can be partitioned into k independent sets.

Definition 1.4.13 (Complete k-Partite Graph)**.** A *complete k-partite graph* is a k-partite graph whose vertices in one partition are adjacent to all vertices of the other partitions. A complete multipartite graph with the partition $(X_1, X_2, \dots, X_k)$ is denoted by $K_{|X_1|,|X_2|,\dots,|X_k|}$.

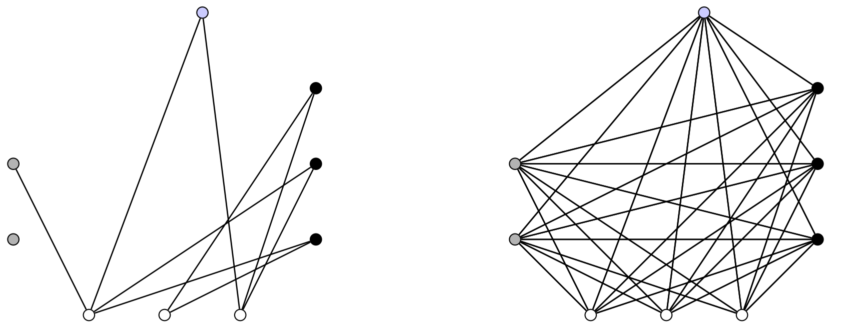

(a) A 4-partite graph (b) A complete 4-partite graph

Figure 1.10: Illustration to multipartite graphs

1.4.4 Regular Graphs

Definition 1.4.14 (Regular Graphs)**.** A graph G is said to be a *regular graph* if all its vertices have the same degree. A graph G is said to be a *k-regular graph* if $d(v) = k \; \forall \; v \in V(G)$. Every complete graph is an $(n-1)$-regular graph.

The degree of all vertices in each partition of a complete bipartite graph is the same. Hence, the complete bipartite graphs are also called *biregular graphs*. Note that, for the complete bipartite graph $K_{|X|,|Y|}$, we have $d_X(v) = |Y|$ and $d_Y(v) = |X|$.

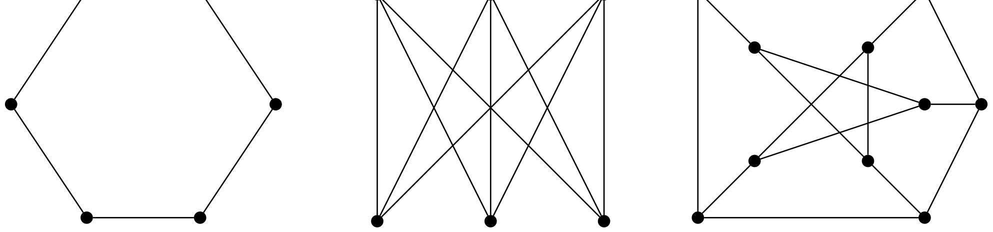

(a) A 2-regular graph (b) A 3-regular graph (c) Petersen Graph

Figure 1.11: Examples of regular graphs

1.5 Isomorphic Graphs

Definition 1.5.1 (Isomorphism of Two Graphs)**.** An *isomorphism* of two graphs G and H is a bijective function $f : V(G) \to V(H)$ such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H.

That is, two graphs G and H are said to be isomorphic if

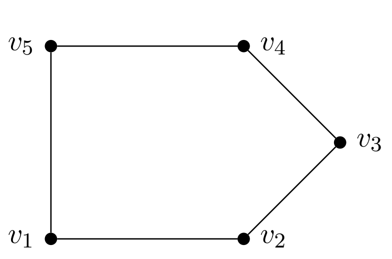
(i) $|V(G)| = |V(H)|$,

(ii) $|E(G)| = |E(H)|$,

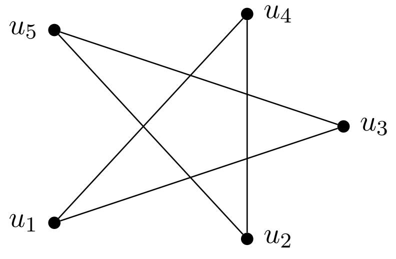
(iii) $v_iv_j \in E(G) \implies f(v_i)f(v_j) \in E(H)$.

This bijection is commonly described as *edge-preserving bijection.*

If an isomorphism exists between two graphs, then the graphs are called *isomorphic graphs* and denoted as $G \simeq H$ or $G \cong H$.
For example, consider the graphs given in Figure 1.12.



(a) G_1.

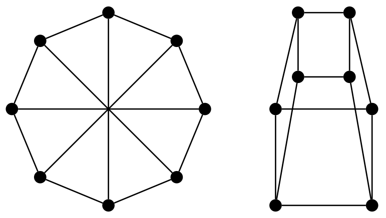


(b) G_2.

Figure 1.12: Examples of isomorphic graphs

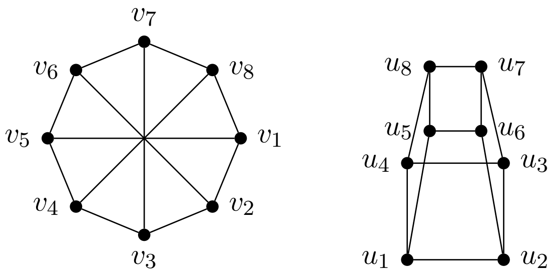
In the above graphs, we can define an isomorphism f from the first graph to the second graph such that $f(v_1) = u_1, f(v_2) = u_3, f(v_3) = u_5, f(v_4) = u_2$ and $f(v_5) = u_4$. Hence, these two graphs are isomorphic.

Problem 1.5.2. Are the two graphs isomorphic? Explain with valid reasons.



Solution: First, label the vertices of the two graphs as shown below:

Define a function f from the first graph to the second such that $f(v_1) = u_1$, $f(v_2) = u_2$, $f(v_3) = u_3$, and $f(v_4) = u_4$. Clearly, we can see that v_1 is adjacent to v_4 in the first graph, but the corresponding vertices u_1 and u_4 are not adjacent in the second graph. So, adjacency is not preserved in the above graphs. Therefore, they are not isomorphic. □



1.6 Graphs and Their Operations

We have already seen that the notion of subgraphs can be defined for any graphs as similar to the definition of subsets to sets under consideration. Similar to the definitions of basic set operations, we can define the corresponding basic operations for graphs also. In addition to these fundamental graph operations, there are some other new and useful operations are also defined on graphs. In this section, we discuss some basic graph operation.

1.6.1 Union, Intersection and Ringsum of Graphs

Definition 1.6.1 (Union of Graphs)**.** The *union* of two graphs G_1 and G_2 is a graph G, written by $G = G_1 \cup G_2$, with vertex set $V(G_1) \cup V(G_2)$ and the edge set $E(G_1) \cup E(G_2)$.

Definition 1.6.2 (Intersection of Graphs)**.** The *intersection* of two graphs G_1 and G_2 is another graph G, written by $G = G_1 \cap G_2$, with vertex set $V(G_1) \cap V(G_2)$ and the edge set $E(G_1) \cap E(G_2)$.

Definition 1.6.3 (Ringsum of Graphs)**.** The *ringsum* of two graphs G_1 and G_2 is another graph G, written by $G = G_1 \oplus G_2$, with vertex set $V(G_1) \cap V(G_2)$ and the edge set $E(G_1) \oplus E(G_2)$, where $\oplus$ is the symmetric difference (XOR Operation) of two sets.

Figure 1.13 illustrates the union, intersection and ringsum of two given graphs.

Remark 1.6.4.
1. *The union, intersection and ringsum operations of graphs are commutative. That is,* $G_1 \cup G_2 = G_2 \cup G_1, G_1 \cap G_2 = G_2 \cap G_1$ *and* $G_1 \oplus G_2 = G_2 \oplus G_1$.
2. *If* G_1 *and* G_2 *are edge-disjoint, then* $G_1 \cap G_2$ *is a null graph, and* $G_1 \oplus G_2 = G_1 \cup G_2$.
3. *If* G_1 *and* G_2 *are vertex-disjoint, then* $G_1 \oplus G_2$ *is empty.*
4. *For any graph* G, $G \cap G = G \cup G$ *and* $G \oplus G$ *is a null graph.*

1.6.2 Complement of Graphs

Definition 1.6.5 (Complement of Graphs)**.** The *complement* or *inverse* of a graph

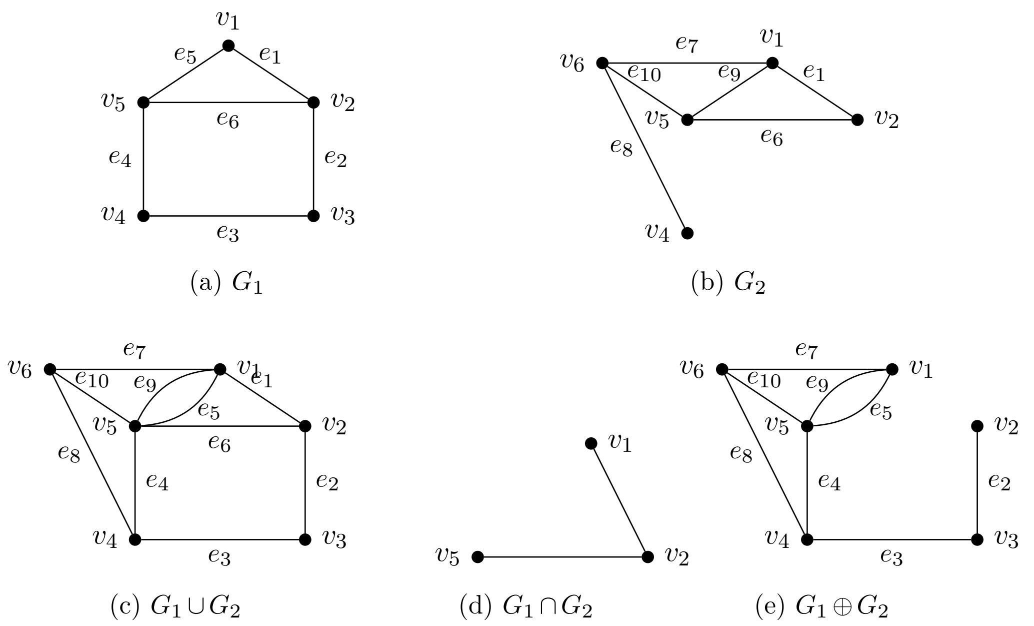


Figure 1.13: Illustrations to graph operations

G, denoted by $\overline{G}$ is a graph with $V(G) = V(\overline{G})$ such that two distinct vertices of $\overline{G}$ are adjacent if and only if they are not adjacent in G.

Remark 1.6.6. *Note that for a graph G and its complement $\overline{G}$, we have*

(i) $G_1 \cup \overline{G} = K_n$;

(ii) $V(G) = V(\overline{G})$;

(iii) $E(G) \cup E(\overline{G}) = E(K_n)$;

(iv) $|E(G)| + |E(\overline{G})| = |E(K_n)| = \binom{n}{2}$.

A graph and its complement are illustrated below.

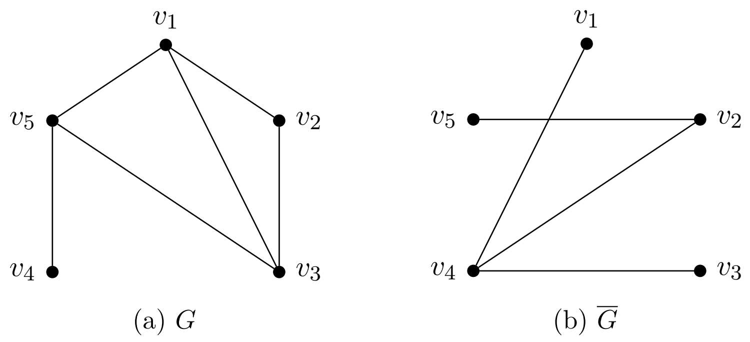


Figure 1.14: A graph and its complement

Theorem 1.6.7. *Two graphs G and H are isomorphic if and only if their complements are isomorphic*

Proof. Let G and H be two isomorphic graphs and let $\overline{G}$ and $\overline{H}$ respectively be their complements. Since G is isomorphic to H, we can define a bijection $f : V(G) \to V(H)$, such that $uv \in E(G)$ if and only if $f(u)f(v) \in V(H)$. Or equivalently, there exists a bijection $f : V(G) \to V(H)$ such that $uv \notin E(G)$ if and only if $f(u)f(v) \in V(H)$.

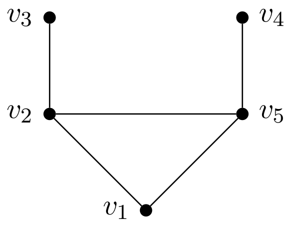
Since the vertex set of G and $\overline{G}$ are the same, the map f is a bijection from $V(\overline{G})$ to $V(\overline{H})$. If $uv \notin E(G)$, then $uv \in \overline{G}$ and in a similar way, if $f(u)f(v) \notin E(H)$, then $f(u)f(v) \in \overline{H}$. Hence $\overline{G}$ and $\overline{H}$.

Conversely, assume that $\overline{G}$ and $\overline{H}$ are isomorphic. Then, by the above paragraph, their complements $\overline{\overline{G}}$ and $\overline{\overline{H}}$ are also isomorphic. That is, $G = \overline{\overline{G}}$ and $H = \overline{\overline{H}}$ are isomorphic. This completes the proof. □

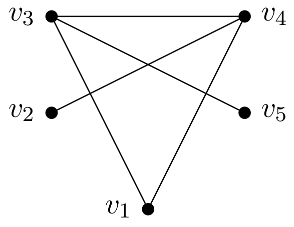
1.6.3 Self-Complementary Graphs

Definition 1.6.8 (Self-Complementary Graphs)**.** A graph G is said to be *self-complementary* if G is isomorphic to its complement. If G is self complementary, then $|E(G)| = |E(\overline{G})| = \frac{1}{2}|E(K_n)| = \frac{1}{2}\binom{n}{2} = \frac{n(n-1)}{4}$.

The following are two examples of self complementary graphs.

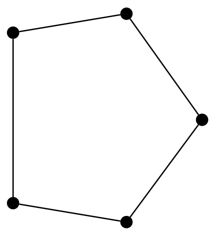


(a) Bull graph G.

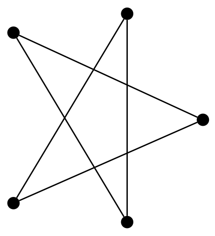


(b) $\overline{G}$.

Figure 1.15: Example of self-complementary graphs



(a) G_2



(b) G_2.

Figure 1.16: Example of self-complementary graphs

Problem 1.6.9. For any self-complementary graph G of order n, show that $n \equiv 0, 1 (\text{mod } 4)$.

Solution: For self-complementary graphs, we have

(i) $V(G) = V(\overline{G})$;
(ii) $|E(G)| + |E(\overline{G})| = \frac{n(n-1)}{2}$;
(iii) $|E(G)| = |E(\overline{G})|$.

Therefore, $|E(G)| = |E(\overline{G})| = \frac{n(n-1)}{4}$. This implies, 4 divides either n or $n-1$. That is, for self-complementary graphs of order n, we have $n \equiv 0, 1 (\text{mod } 4)$. □

(Note that we say $a \equiv b (\text{mod } n)$, which is read as "$a$ is congruent to b modulo n", if $a - b$ is completely divisible by n).

Theorem 1.6.10. *A graph G is regular if and only if its complement is regular.*

Proof. Let G be a k-regular graph on n vertices and let H be its complement. Therefore, $G \cup H = K_n$. Thus, for any vertex $v \in V(G)$, we have

$$\begin{aligned} d_G(v) + d_H(v) &= n-1 \\ k + d_H(v) &= n-1 \\ d_H(v) &= n-1-k. \end{aligned}$$

That is, H is an $(n-k-1)$-regular graph. Conversely, assume that $d_H(v) = \ell$ for all $v \in H$. Therefore, we have

$$\begin{aligned} d_H(v) + d_{\overline{H}}(v) &= n-1 \\ \ell + d_{\overline{H}}(v) &= n-1 \\ d_{\overline{H}}(v) &= n-1-\ell. \end{aligned}$$

Since $d_{\overline{H}} = G$, we have G is $(n-\ell-1)$-regular graph, completing the proof. □

Problem 1.6.11. If a graph and its complement are r-regular, then show that G has odd number of vertices.

Solution: Let G be an r-regular graph and H be its complement which is also r-regular. Then, we have

$$\begin{aligned} d_G(v) + d_H(v) &= n-1 \\ r + r &= n-1 \\ 2r &= n-1 \\ \therefore n &= 2r+1. \end{aligned}$$

Thus, G has odd number of vertices. □

1.6.4 Join of Graphs

Definition 1.6.12. The *join* of two graphs G and H, denoted by $G+H$ is the graph such that $V(G+H) = V(G) \cup V(H)$ and $E(G+H) = E(G) \cup E(H) \cup \{xy : x \in V(G), y \in V(H)\}$.

In other words, the join of two graphs G and H is defined as the graph in which every edge of the first graph is adjacent to all vertices of the second graph.

Figure 1.17 illustrates the join of two graphs P_3 and P_4 and Figure 1.18 illustrates the join of two graphs C_5 and P_2.

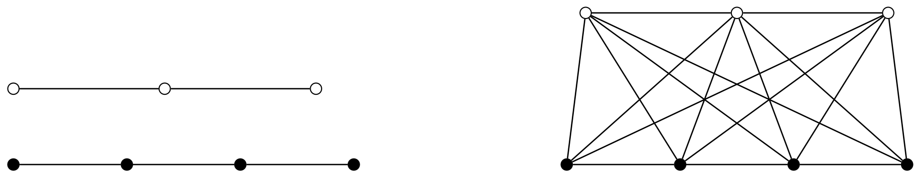


Figure 1.17: The join of the paths P_4 and P_3.

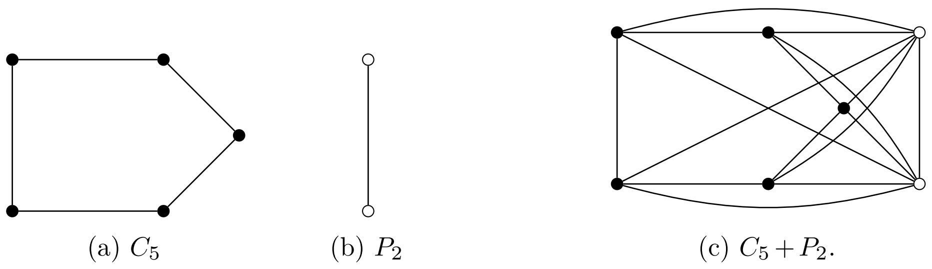

(a) C_5 (b) P_2 (c) $C_5 + P_2$.

Figure 1.18: The join of the cycle C_5 and the path P_2.

1.6.5 Deletion and Fusion

Definition 1.6.13 (Edge Deletion in Graphs)**.** If e is an edge of G, then $G-e$ is the graph obtained by removing the edge of G. The subgraph of G thus obtained is called an *edge-deleted subgraph* of G. Clearly, $G-e$ is a spanning subgraph of G.

Similarly, vertex-deleted subgraph of a graph is defined as follows:

Definition 1.6.14 (Vertex Deletion in Graphs)**.** If v is a vertex of G, then $G-v$ is the graph obtained by removing the vertex v and all edges G that are incident on v. The subgraph of G thus obtained is called an *vertex-deleted subgraph* of G. Clearly, $G-v$ will not be a spanning subgraph of G.

Figure 1.19 illustrates the edge deletion and the vertex deletion of a graph G.

Definition 1.6.15 (Fusion of Vertices)**.** A pair of vertices u and v are said to be *fused* (or *merged* or *identified*) together if the two vertices are together replaced

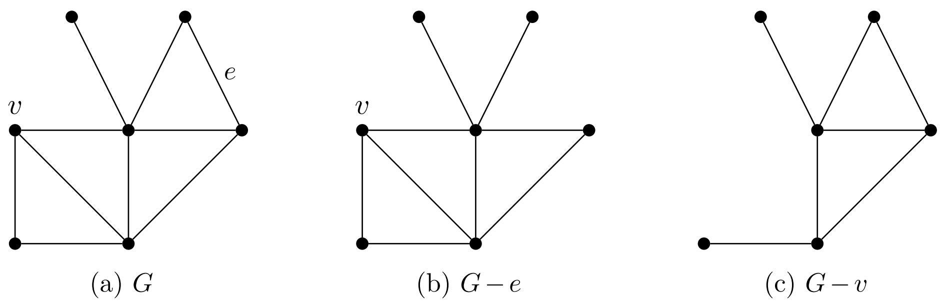


Figure 1.19: Illustrations to edge deletion and vertex deletion

by a single vertex w such that every edge incident with either u or v is incident with the new vertex w (see Figure 1.20).

Note that the fusion of two vertices does not alter the number of edges, but reduces the number of vertices by 1.

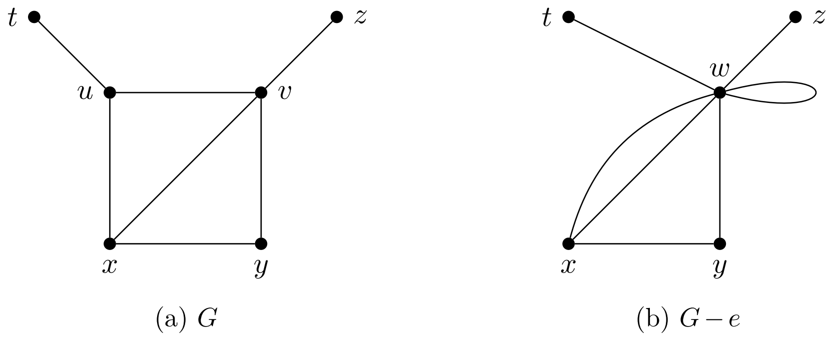


Figure 1.20: Illustrations to fusion of two vertices

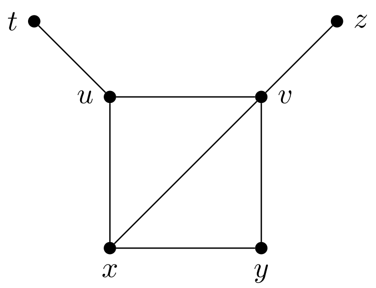
1.6.6 Edge Contraction

Definition 1.6.16 (Edge Contraction in Graphs)**.** An *edge contraction* of a graph G is an operation which removes an edge from a graph while simultaneously merging its two end vertices that it previously joined. Vertex fusion is a less restrictive form of this operation.

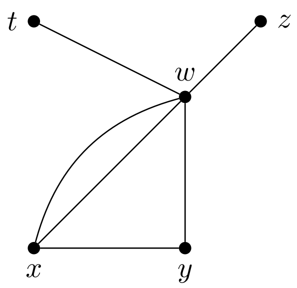
A graph obtained by contracting an edge e of a graph G is denoted by $G \circ e$.

1.6.7 Subdivision of a Graph

Definition 1.6.17 (Subdivision of an Edge)**.** Let $e = uv$ be an arbitrary edge in G. The *subdivision* of the edge e yields a path of length 2 with end vertices u and v with a new internal vertex w (That is, the edge $e = uv$ is replaced by two new edges, uw and wv).

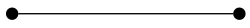


(a) G

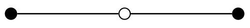


(b) $G \circ uv$

Figure 1.21: Illustrations to edge contraction of a graph.



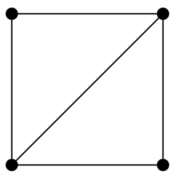
(a) G



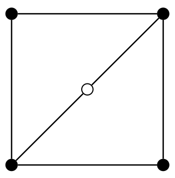
(b) G

Figure 1.22: Subdivision of an edge

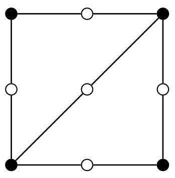
Definition 1.6.18 (Subdivision of a Graph)**.** A *subdivision* of a graph G (also known as an *expansion* of G) is a graph resulting from the subdivision of (some or all) edges in G (see 1.23). The newly introduced vertices in the subdivisions are represented by white vertices.



(a) G



(b)



(c)

Figure 1.23: Illustrations to Subdivision of graphs

Definition 1.6.19 (Homeomorphic Graphs)**.** Two graphs are said to be *homeomorphic* if both can be obtained by the same graph by subdivisions of edges.

In Figure 1.23, the second and third graphs are homeomorphic, as they are obtained by subdividing the edges of the first graph in the figure.

1.6.8 Smoothing a Vertex

Definition 1.6.20 (Smoothing Vertices in Graphs)**.** The reverse operation, *smoothing out* or *smoothing* a vertex w of degree 2 with regards to the pair of edges (e_i, e_j) incident on w, removes w and replaces the pair of edges (e_i, e_j) containing w with a new edge e that connects the other endpoints of the pair (e_i, e_j) (see the illustration).

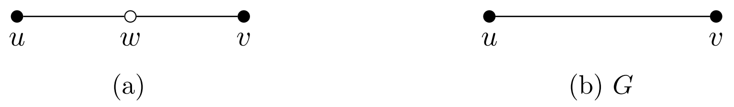


Figure 1.24: Smoothing of the vertex w

Smoothing of a vertex of a graph G is also called an elementary transformation of G.

Problem 1.6.21. *Show that a graph obtained by subdividing all edges of a graph G is a bipartite graph.*

Solution: Let H be the subdivision of G. Let $X = V(G)$ and Y be the newly introduced vertices during subdivision. Clearly, $X \cup Y = V(H)$. Note that adjacency is not defined among the vertices of Y. When we subdivide an edge uv in G, then the edge uv will be removed and hence u and v becomes non-adjacent in H. Therefore, no two vertices in X can be adjacent in H. Thus, H is bipartite. □

1.7 Decomposition and Factorisation of Graphs

Definition 1.7.1 (Decomposition of a Graph)**.** A graph G is said to be *decomposed* into two subgraphs $G_1, G_2, \dots G_k$, if $\bigcup_{i=1}^{k} G_i = G$ and $E(G_i) \cap E(G_j) = \emptyset$, where $i \neq j$.

Definition 1.7.2 (H-Decomposition of a Graph)**.** An *H-decomposition* of a graph G is a decomposition of G into subgraphs $G_1, G_2, \dots, G_k$, where each subgraph G_i is isomorphic to H.

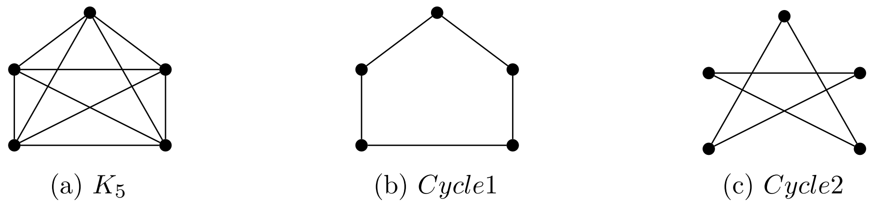


Figure 1.25: Decomposition of a K_5 into cycles

Definition 1.7.3 (Factorisation of a Graph)**.** An *factorisation* of a graph G is a decomposition of G into subgraphs $G_1, G_2, \dots, G_k$, where each subgraph G_i is a spanning subgraph of G.

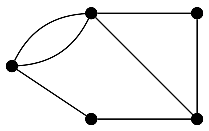
Definition 1.7.4 (r-Factorisation of a Graph)**.** An *r-factor* of G is an r-regular spanning subgraph of G. Thus, an *r-factorisation* of G is a decomposition of G into r-factors.

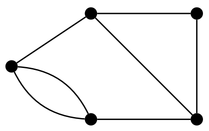
Figure 1.25 is an illustration to a 2-factorisation of the complete graph K_5.

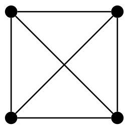
1.8 Exercise Problems

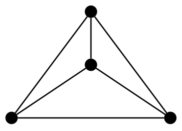
1. Show that every loop-less graph G has a bipartite subgraph with at least $\frac{\epsilon}{2}$ edges.
2. Verify whether graph isomorphism is an equivalence relation?
3. For $k > 0$, show that a k-regular bipartite graph has the same number of vertices in each of its partite sets.
4. Show that every simple graph on n vertices subgraph of K_n.
5. Show that every subgraph of a bipartite graph is bipartite.
6. Verify whether the integer sequences $(7,6,5,4,3,3,2)$ and $(6,6,5,4,3,3,1)$ are graphical.
7. Show that if G is simple and connected but not complete, then G has three vertices u, v and w such that $uv, vw \in E(G)$, but $uw \notin E$.
8. Show that every induced subgraph of a complete graph K_n is also a complete subgraph.
9. If G is an r-regular graph, then show that r divides the size of G.
10. Show that every subgraph of a bipartite graph is bipartite.
11. If G is an r-regular graph and r is odd, then show that $\frac{\epsilon}{r}$ is an even integer.
12. Let G be a graph in which there is no pair of adjacent edges. What can you say about the degree of the vertices in G?
13. Let G be a graph with n vertices and e edges and let m be the smallest positive integer such that $m \geq \frac{2e}{n}$. Prove that G has a vertex of degree at least m.
14. Prove that it is impossible to have a group of nine people at a party such that each one knows exactly five of the others in the group.
15. Let G be a graph with n vertices, t of which have degree k and the others have degree $k+1$. Prove that $t = (k+1)n - 2e$, where e is the number of edges in G.
16. Let G be a k-regular graph, where k is an odd number. Prove that the number of edges in G is a multiple of k.
17. Let G be a graph with n vertices and exactly $n-1$ edges. Prove that G has either a vertex of degree 1 or an isolated vertex.
18. What is the smallest integer n such that the complete K_n has at least 500 edges?
19. Prove that there is no simple graph with six vertices, one of which has degree 2, two have degree 3, three have degree 4 and the remaining vertex has degree 5.
20. Prove that there is no simple graph on four vertices, three of which have degree 3 and the remaining vertex has degree 1.
21. Let G be a simple regular graph with n vertices and 24 edges. Find all possible values of n and give examples of G in each case.
22. Show that the complement of a complete bipartite graph is the disjoint union of two complete graphs.
23. The isomorphic image of a graph walk W is a walk of the same length.

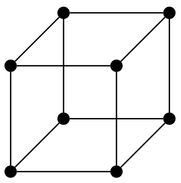
24. For any graphs G and H, the ringsum $G \oplus H$ is empty if and only if $E(G) = E(H)$.
25. Show that the ringsum of two edge-disjoint collections of cycles is a collection of cycles.
26. For any graph G with six vertices, then G or its complement $\overline{G}$ contains a triangle.
27. Every graph G contains a bipartite spanning subgraph whose size is at least half the size of G.
28. Any graph G has a regular supergraph H of degree $\Delta(G)$ such that G is an induced subgraph of H.
29. how that if a self-complementary graph contains pendent vertex, then it must have at least another pendent vertex.
30. Draw all the non-isomorphic self complementary graphs on four vertices.
31. Prove that a graph with n vertices ($n > 2$) cannot be bipartite if it has more than $\frac{n^2}{4}$ edges.
32. Verify whether the join of two bipartite graphs is bipartite. Justify your answer.
33. What is the order and size of the join of two graphs?
34. Does the join of two graphs hold commutativity? Illustrate with examples.
35. Construct a simple connected r-regular graph on 6 vertices and a simple connected s-regular graph on 7 vertices for all possible values of r and s.
36. Let G be a graph of order $2n+1$ such that the degree of every vertex of G is either $n+1$ or $n+2$. Then, show that G has at least $n+1$ vertices of degree $n+2$and at least $n+2$ vertices of degree $n+1$.
37. Two graphs G and H have the same order, size and degree sequence. Are they necessarily isomorphic? Justify your comment with suitable illustrations.
38. The degree of each vertex of a certain graph of order 12 and size 31 is either 4 or 6. Find the number of vertices falling in each category. For any simple connected graph G of order n and size m, show that $n-1 \leq m \leq \frac{n(n-1)}{2}$.
39. Check whether the following pairs of graphs are isomorphic? Justify your answer.

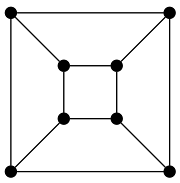

(a) G

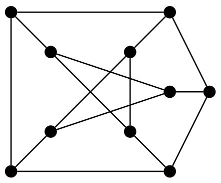

(b) H

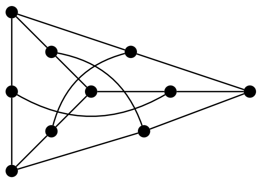

(a) G

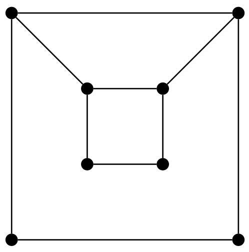

(b) H

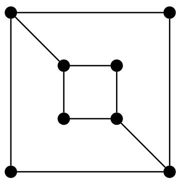

(a) G

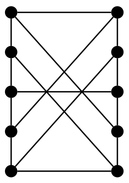

(b) H

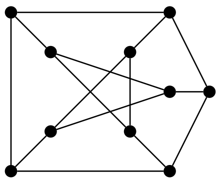

(a) G


(b) H


(a) G


(b) H


(a) G


(b) H

****

Chapter 2

Connectivity in Graphs

****

2.1 Paths, Cycles and Distances in Graphs

Definition 2.1.1 (Walks)**.** A *walk* in a graph G is an alternating sequence of vertices and connecting edges in G. In other words, a *walk* is any route through a graph from vertex to vertex along edges. If the starting and end vertices of a walk are the same, then such a trail is called a *closed walk*.

A walk can end on the same vertex on which it began or on a different vertex. A walk can travel over any edge and any vertex any number of times.

Definition 2.1.2 (Trails and Tours)**.** A *trail* is a walk that does not pass over the same edge twice. A trail might visit the same vertex twice, but only if it comes and goes from a different edge each time. A *tour* or a *closed trail* is a trail that begins and ends on the same vertex.

Definition 2.1.3 (Paths and Cycles)**.** A *path* is a walk that does not include any vertex twice, except that its first vertex might be the same as its last. A *cycle* or a *circuit* is a path that begins and ends on the same vertex.

Definition 2.1.4 (Length of Paths and Cycles)**.** The *length* of a walk or trail or path or cycle is the number of edges in it.

A path of order n is denoted by P_n and a cycle of order n is denoted by C_n. Every edge of G can be considered as a path of length 1. Note that the length of a path on n vertices is $n-1$.

A cycle having odd length is usually called an *odd cycle* and a cycle having even length is called an *even cycle*.

Definition 2.1.5 (Distance between two vertices)**.** The *distance* between two vertices u and v in a graph G, denoted by $d_G(u,v)$ or simply $d(u,v)$, is the length (number of edges) of a *shortest path* (also called a *graph geodesic*) connecting them. This distance is also known as the *geodesic distance*.

Definition 2.1.6 (Eccentricity of a Vertex)**.** The *eccentricity* of a vertex v, denoted by $\varepsilon(v)$, is the greatest geodesic distance between v and any other vertex. It can be thought of as how far a vertex is from the vertex most distant from it in the graph.

Definition 2.1.7 (Radius of a Graph)**.** The *radius* r of a graph G, denoted by $rad(G)$, is the minimum eccentricity of any vertex in the graph. That is, $rad(G) = \min\limits_{v \in V(G)} \varepsilon(v)$. In other words, $rad(G) = \min\limits_{u \in V(G)} \max\limits_{v \in V(G)} \{d(u,v)\}$.

Definition 2.1.8 (Diameter of a Graph)**.** The *diameter* of a graph G, denoted by $diam(G)$ is the maximum eccentricity of any vertex in the graph. That is, $diam(G) = \max\limits_{v \in V(G)} \varepsilon(v)$. In other words, $diam(G) = \max\limits_{u \in V(G)} \max\limits_{v \in V(G)} \{d(u,v)\}$.

Here, note that the diameter of a graph need not be twice its radius unlike in geometry. We can even see many graphs having same radius and diameter. Complete graphs are examples of the graphs with radius equals to diameter.

Definition 2.1.9 (Center of a Graph)**.** A *center* of a graph G is a vertex of G whose eccentricity equal to the radius of G.

Definition 2.1.10 (Peripheral Vertex of a Graph)**.** A *peripheral vertex* in a graph of diameter d is one that is distance d from some other vertex. That is, a peripheral vertex is a a vertex that achieves the diameter. More formally, a vertex v of G is peripheral vertex of a graph G, if $\varepsilon(v) = d$.

Unlike geometry, there is no direct relation between the radius and the diameter of a graph. Interestingly, we can find the graphs with their radii are equal to the diameters. A complete graph is such a graph where the radius and the diameter of K_n is 1 for any $n \in \mathbb{N}$. But for a complete bipartite graph $K_{a,b}$, the radius is 1 while the diameter is 2.

Problem 2.1.11. Find its center and diameter of the following graph.

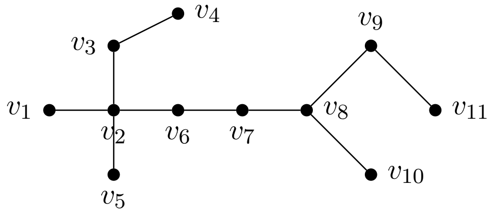


Figure 2.1: G

Solution: The eccentricities of the vertices of the given graph are given in the following table.

v	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	ε
v_1	0	1	2	3	2	2	3	4	5	5	6	6
v_2	1	0	1	2	1	1	2	3	4	4	5	5
v_3	2	1	0	1	1	3	4	5	5	5	6	6
v_4	3	2	1	0	3	3	4	5	5	6	7	7
v_5	2	1	2	3	0	2	3	4	5	5	6	6
v_6	2	1	2	3	2	0	1	2	3	3	4	4
v_7	3	2	3	4	3	1	0	1	2	2	3	4
v_8	4	3	4	5	4	2	1	0	1	1	2	5
v_9	5	4	5	6	5	3	2	1	0	2	1	6
v_{10}	5	4	5	6	5	3	2	1	0	2	1	6
v_{11}	6	5	6	7	6	4	3	2	1	0	3	7

Table 2.1: Eccentricities of vertices of the graph in Figure 2.1.

Form the above table, we have diameter of the graph is 7 and radius of the graph is 4. Therefore, the vertices v_6 and v_7 form the center of G. □

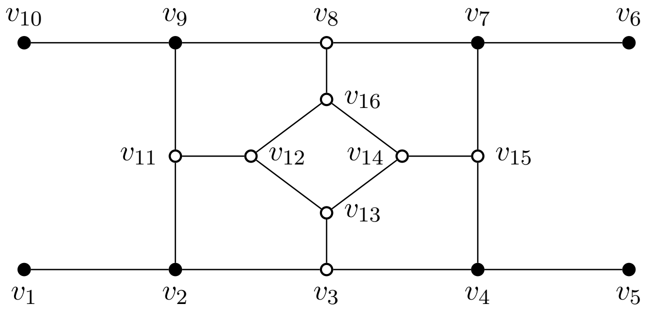


Figure 2.2: A graph with eight centers.

The distances between vertices in the above graph are given in Table 2.2. Note that a vertex v_i is represented by i in the table (to save the space).

v	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	ε
1	0	1	2	3	4	6	5	4	3	4	2	3	3	4	4	4	6
2	1	0	1	2	3	5	4	3	2	3	1	2	2	3	3	3	5
3	2	1	0	1	2	4	3	4	3	4	2	2	1	1	2	3	4
4	3	2	1	0	1	3	2	3	4	5	5	3	2	2	1	3	5
5	4	3	2	1	0	4	3	4	5	6	4	4	3	3	2	4	6
6	6	5	4	3	4	0	1	2	3	4	4	4	4	3	2	3	6
7	5	4	3	2	3	1	0	1	2	3	3	3	3	2	1	2	5
8	4	3	4	3	4	2	1	0	1	2	2	2	3	2	2	1	4
9	3	2	3	4	5	3	2	1	0	1	1	2	3	3	3	2	5
10	4	3	4	5	6	4	3	2	1	0	2	3	4	4	4	3	6
11	2	1	2	3	4	4	3	2	1	1	0	1	2	3	4	2	4
12	3	2	2	3	4	4	2	1	2	3	1	0	1	2	3	1	4
13	3	2	1	2	3	4	3	3	3	4	2	1	0	1	2	2	4
14	4	3	2	2	3	3	2	2	3	4	3	2	1	0	1	1	4
15	4	3	2	1	2	2	1	2	3	4	4	3	2	1	0	2	4
16	4	3	3	3	4	3	2	1	2	3	2	3	2	1	2	0	4

Table 2.2: Eccentricities of vertices of the graph in Figure 2.2.

Note that the radius of G is given by $r(G) = \min\{\varepsilon(v)\} = 4$ and the diameter of G is given by $diam(G) = \max\{\varepsilon(v)\} = 6$ and all eight central vertices are represented by white vertices in Figure 2.2.

Theorem 2.1.12. *For any graph G, $rad(G) \leq diam(G) \leq 2rad(G)$.*

Proof. We know that

$$diam(G) = \max_{u \in V(G)} \max_{v \in V(G)} \{d(u,v)\}$$

and

$$rad(G) = \min_{u \in V(G)} \max_{v \in V(G)} \{d(u,v)\}$$

By the definition, it is clear that

$$rad(G) \leq diam(G) \tag{2.1}$$

Let x be a central vertex (that is, of minimal distance from all the vertices) of G. Then, we have $d(u,x) \leq rad(G)$ and $d(x,v) \leq rad(G)$, for any two vertices $u,v \in V(G)$. Adding these two inequalities, we get $d(u,v) \leq d(u,x) + d(x,v) \leq 2rad(G)$. Since u and v are arbitrary, it is clear that the inequality holds for any two farthest vertices also. Therefore,

$$diam(G) \leq 2rad(G) \tag{2.2}$$

Therefore, by Inequality (2.1) and Inequality (2.2), we have $rad(G) \leq diam(G) \leq 2rad(G)$. □

Note that for a general graph, there may be several centers and a center need not necessarily be on a diameter.

Definition 2.1.13 (Geodetic Graph)**.** A graph in which any two vertices are connected by a unique shortest path is called a *geodetic graph.*

Theorem 2.1.14. *If G is a simple graph with $diam(G) \geq 3$, then $diam(\overline{G}) \leq 3$.*

Proof. If $diam(G) \geq 3$, then there exist at least two non-adjacent vertices u and v in G such that u and v have no common neighbours in G. Hence, every vertex x in $G-\{u,v\}$ is non-adjacent to u or v or both in G. This makes x adjacent to u or v or both in $\overline{G}$. Moreover, $uv \in E(\overline{G})$. So, for every pair of vertices x,y, there is an x,y path of length at most 3 in $\overline{G}$ through the edge uv. Hence, $diam(\overline{G}) \leq 3$. □

2.2 Connected Graphs

Definition 2.2.1 (Connectedness in a Graph)**.** Two vertices u and v are said to be *connected* if there exists a path between them. If there is a path between two vertices u and v, then u is said to be *reachable* from v and vice versa. A graph G is said to be *connected* if there exist paths between any two vertices in G.

Definition 2.2.2 (Component of a Graph)**.** A *connected component* or simply, a *component* of a graph G is a maximal connected subgraph of G.

Each vertex belongs to exactly one connected component, as does each edge. A connected graph has only one component.

A graph having more than one component is a *disconnected graph* (In other words, a disconnected graph is a graph which is not connected). The number of components of a graph G is denoted by $\omega(G)$.

In view of the above notions, the following theorem characterises bipartite graphs.

Theorem 2.2.3. *A connected graph G is bipartite if and only if G has no odd cycles.*

Proof. Suppose that G is a bipartite graph with bipartition (X,Y). Assume for contradiction that there exists a cycle $v_1,v_2,v_3,\ldots\ldots,v_k,v_1$ in G with k odd. Without loss of generality, we may additionally assume that $v_1 \in X$. Since G is bipartite, $v_2 \in Y$, $v_3 \in X$, $v_4 \in Y$ and so on. That is, $v_i \in X$ for odd values of i and $v_i \in Y$ for even values of i. Therefore, $v_k \in X$. But, then the edge $v_k,v_1 \in E$ is an edge with both endpoints in X, which contradicts the fact that G is bipartite. Hence, a bipartite graph G has no odd cycles.

Conversely, assume that G is a graph with no odd cycles. Let $d(u,v)$ denote the distance between two vertices u and v in G. Pick an arbitrary vertex $u \in V$ and define $X = \{x \in V(G) : d(x,u) \text{ is even}\}$. Clearly, $u \in X$ as $d(u,u) = 0$. Now, define another $Y = \{y \in V(G) : d(u,y) \text{ is odd}\}$. That is, $Y = V - X$. If possible, assume that there exists an edge $vw \in E(G)$ such that $v,w \in X$ (or $v,w \in Y$). Then, by construction $d(u,v)$ and $d(u,w)$ are both even (or odd). Let $P(u,w)$ and $P(u,v)$ be the shortest paths connecting u to w, and u to v respectively. Then, the cycle given by $P(u,w) \cup \{wv\} \cup P(v,u)$ has odd length $1+d(u,w)+d(u,v)$, which is a contradiction. Therefore, no such edge wv may exist and G is bipartite. □

Corollary 2.2.4. *Any non-trivial subgraph of a bipartite graph is also bipartite.*

Proof. Let G be a bipartite graph and let H be a non-trivial subgraph of G. If possible, let H is not bipartite. Then, by Theorem 2.2.3, H contains at least one odd cycle, say C. Since H is a subgraph of G, the odd cycle C is cycle in G also, contradicting to the fact that G is bipartite. Therefore, H must be bipartite. □

Theorem 2.2.5. *A graph G is disconnected if and only if its vertex set V can be partitioned into two non-empty, disjoint subsets V_1 and V_2 such that there exists no edge in G whose one end vertex is in subset V_1 and the other in the subset V_2.*

Proof. Suppose that such a partitioning exists. Consider two arbitrary vertices u and v of G, such that $u \in V_1$ and $v \in V_2$. No path can exist between vertices u and v; otherwise, there would be at least one edge whose one end vertex would be in V_1 and the other in V_2. Hence, if a partition exists, G is not connected.

Conversely, assume that G is a disconnected graph. Consider a vertex u in G. Let V_1 be the set of all vertices that are joined by paths to u. Since G is disconnected, V_1 does not include all vertices of G. The remaining vertices will form a (nonempty) set V_2. No vertex in V_1 is joined to any vertex in V_2 by an edge. Hence, we get the required partition. □

Theorem 2.2.6. *If a graph has exactly has two vertices of odd degree, then there exists a path joining these two vertices.*

Proof. Let G be a graph with two vertices v_1 and v_2 of odd degree and all other vertices of even degree. Then, by Theorem 1.2.8, both of them should lie in the same component of G. Since every component of G must be connected, there must be a path between v_1 and v_2. □

Theorem 2.2.7. *For $k \geq 2$, a k-regular bipartite graph has no cut-edges.*

Proof. For $k \geq 2$, let G be a k-regular bipartite graph. If possible, let an edge $e = xy$ be a cut-edge of G. Let $H = G - e$. Thus, we have $d_H(x) = d_H(y) = k-1$ and for any other vertex v, we have $d_H(v) = k$. Since e is a cut-edge of G, H is disconnected and has two components say H_1 and H_2. Without loss of generality, let $x \in V(H_1)$ and $y \in V(H_2)$.

Since graph G is bipartite, the graph H and hence H_1 and H_2 are also both bipartite. Let (X_1, Y_1) be the bipartition of H_1 and let $x \in X_1$. Therefore,

$$|E(H_1)| = \sum_{v \in X_1} d(v) = \sum_{v \in Y_1} d(v)$$

$$= |X|k-1 = \quad |Y|k$$

which is possible only when $k=1$, a contradiction to the choice of k. Thus, our initial assumption that e is a cut-edge was wrong. Therefore, G has no cut-edge. □

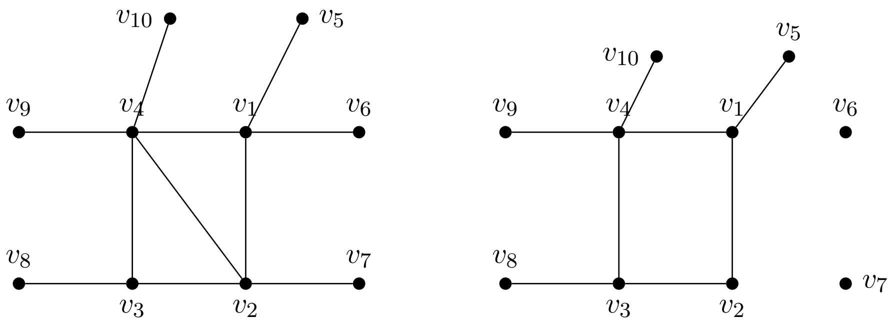
Problem 2.2.8. If a disconnected graph G with two components has two odd degree vertices, show that both of them will be in the same component.

Solution: Assume that G is a disconnected graph with two components, say H_1 and H_2. Given that G has exactly two vertices of odd degree, say u and v. If possible, let $u \in H_1$ and $v \in H_2$. Since H_1 and H_2 are connected graphs by themselves and the sum of degrees of vertices in H_1 (and in H_2) is odd, which is a contradiction to Theorem 1.2.6. Therefore, either $u,v \in H_1$ or $u,v \in H_2$.

2.3 Edge Deleted and Vertex Deleted Subgraphs

Definition 2.3.1 (Edge Deleted Subgraphs)**.** Let $G(V,E)$ be a graph and $F \subseteq E$ be a set of edges of G. Then, the graph obtained by deleting F from G, denoted by $G-F$, is the subgraph of G obtained from G by removing all edges in F. Note that $V(G-F) = V(G)$. That is, $G-F = (V, E-F)$.

Note that any edge deleted subgraph of a graph G is a spanning subgraph of G.



(a) G (b) $G' = G - \{v_1v_6, v_2v_4, v_2v_7\}$.

Figure 2.3: A graph and its edge deleted subgraph.

Definition 2.3.2 (Vertex Deleted Subgraphs)**.** Let $W \subseteq V(G)$ be a set of vertices of G. Then the graph obtained by deleting W from G, denoted by $G-W$, is the subgraph of G obtained from G by removing all vertices in W and all edges incident to those vertices.

See Figure 2.4 for illustration of a vertex-deleted subgraph of a given graph.

Problem 2.3.3. Prove that a graph G with n vertices is connected, if it has more than $\frac{(n-1)(n-2)}{2}$ edges.

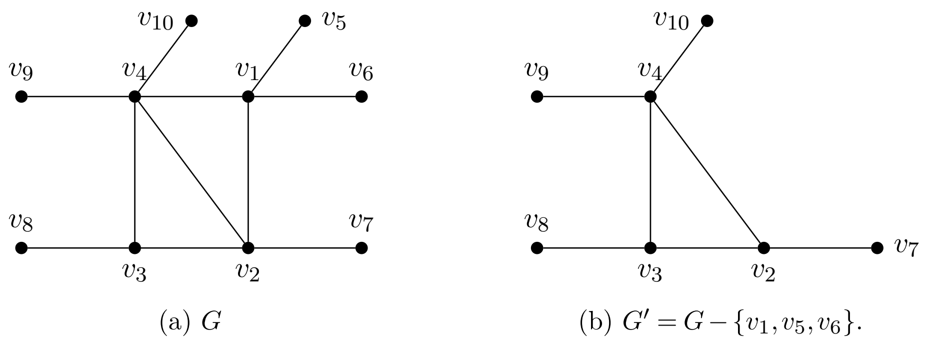


Figure 2.4: A graph and its edge deleted subgraph.

Solution: Consider the complete graph K_n. The minimum number of edges to be removed from K_n so that it becomes disconnected, is $n-1$. In this case, the reduced graph has two components – a complete graph K_{n-1} and an isolated vertex. Moreover, this reduced graph has $\frac{n(n-1)}{2} - (n-1) = \frac{(n-1)(n-2)}{2}$ edges. That is, the highest number of edges a disconnected graph on n vertices can have without being conected is $\frac{(n-1)(n-2)}{2}$. Therefore, any graph on n vertices with more than $\frac{(n-1)(n-2)}{2}$ edges is connected. □

Cut-Edges and Cut-Vertices

Definition 2.3.4 (Cut-Edge). An edge e of a graph G is said to be a *cut-edge* or a *bridge* of G if $G - e$ is disconnected.

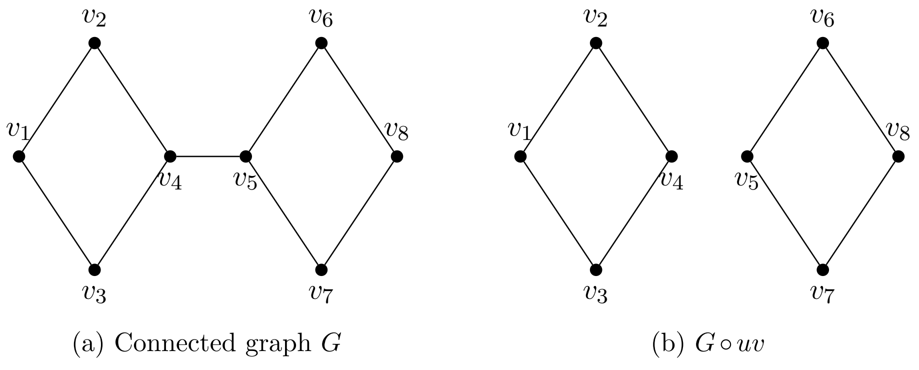


Figure 2.5: Disconnected graph $G - v_4v_5$

In the above graph G, the edge v_4v_5 is a cut-edge, since $G - v_4v_5$ is a disconnected graph.

The following is a necessary and sufficient condition for an edge of a graph G to be a cut edge of G.

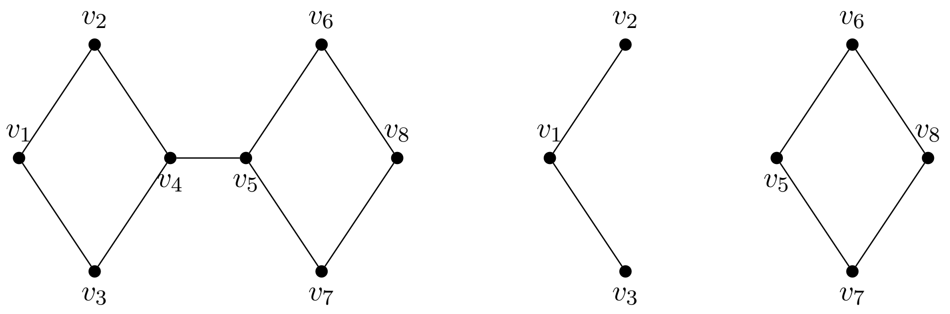
Theorem 2.3.5. *An edge e of a graph G is a cut-edge of G if and only if it is not contained in any cycle of G.*

Proof. Let $e = uv$ be a cut edge of G. Then, the vertices u and v must be in different components of $G - e$. If possible, let e is contained in cycle C in G. Then, $C - e$ is a path between u and v in $G - e$, a contradiction to the fact that u and v are in different components of $G - e$. Therefore, e can not be in any cycle of G.

Conversely, assume that e is not in any cycle of G. Then, there is no (u, v)-path other than e. Therefore, u and v are in different components of $G - e$. That is, $G - e$ is disconnected and hence e is a cut-edge of G. $\square$

Definition 2.3.6 (Cut-Vertex)**.** A vertex v of a graph G is said to be a *cut-vertex* of G if $G - v$ is disconnected.

A cut-vertex is also called a *cut-node* or an *articulation point*.



(a) Connected graph G (b) $G - v_4$

Figure 2.6: disconnected graph $G - v_4$

In graph G, v_4 is a cut-vertex as $G - v_4$ is a disconnected graph. Similarly, v_5 is also a cut-vertex of G.

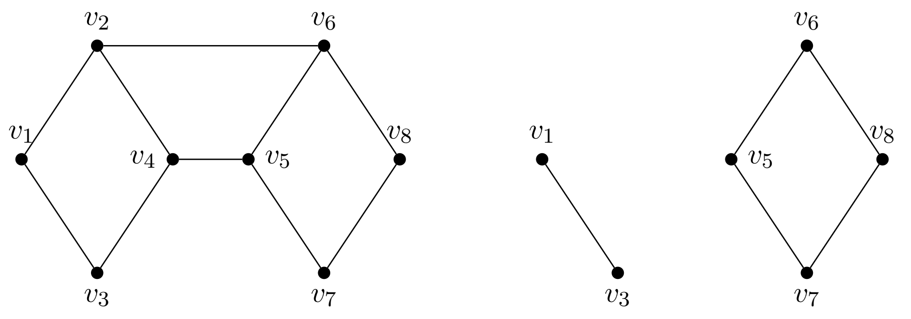
Since removal of any pendent vertex will not disconnect a given graph, every cut-vertex will have degree greater than or equal to 2. But, note that every vertex v, with $d(v) \geq 2$ need not be a cut-vertex.

2.4 Vertex-Cut of a Graph

Definition 2.4.1 (Vertex-Cut)**.** A subset W of the vertex set V of a graph G is said to be a *vertex-cut* or a *Separating Set* of G if $G - W$ is disconnected. In the above graph, the vertex set $W = \{v_3, v_4\}$ is a vertex-cut of G.

The following graph illustrates a vertex-cut of a graph.

Remark 2.4.2. If v is a cut-vertex of G, then $\{v\}$ is a vertex-cut of G. Any set of cut vertices in G is also a vertex-cut of G.



(a) Connected graph G (b) Disconnected graph $G - \{v_2, v_4\}$

Figure 2.7: disconnected graph $G - \{v_4v_5, v_2v_6, v_3v_7\}$.

The following result is a necessary and sufficient condition for a vertex of a graph G to be a cut-vertex of G.

Theorem 2.4.3. *A vertex v of connected graph G is a cut-vertex of G if and only if there exist two (or more) edges incident on v such that no cycles in G contain both (all) edges simultaneously.*

Proof. Let v be an arbitrary vertex of a graph G and let $e_1 = uv$ and $e_2 = vw$ be two edges incident on v.

Assume v be cut-vertex of G. If possible, let e_1 and e_2 lie on the same cycle, say C. Then, the path uvw and the path $C - v$ are two distinct uw-paths in G. Hence, u and w lie in the same component of $G - v$, which is a contradiction to the assumption that v is a cut-vertex of G. Therefore, C contains any one of these two edges.

Conversely, assume that no cycle in G contains both edges e_1 and e_2 incident on v. Then, uvw is the only uw-path in G and hence u and w lie on two distinct components in $G - v$. Then, v is a cut-vertex of G, completing the proof. □

The following result is another necessary and sufficient condition for a vertex of a graph G to be a cut-vertex of G.

Theorem 2.4.4. *A vertex v of connected graph G is a cut-vertex of G if and only if there exist two vertices, say x and y, in G such that every uw-path passes through v.*

Proof. Let v, x, y be three vertices of a graph G such that every xy-paths in G pass through v. Then, x and y are not connected in $G - v$. That is, $G - v$ is disconnected and hence v is a cut edge of G.

Let v be a cut vertex of a graph G. Let x and y be two vertices such that they lie on two components in $G - v$. If possible, let some xy-paths pass through v and some other do not. Let P be an xy-path which does not pass through v.

Pick two adjacent vertices u and w of v such that they lie on different components of $G - v$. Without loss of generality, let u and x are in the same component, say K_1, of $G-v$, while w and y are in the other component, say K_2, of $G-v$. Since K_1 is connected, there exists a path, say P_1, between u and x. Similarly, there exists a path P_2 between w and y in K_2. Then, we note that $P_1 \cup P \cup P_2 \cup uvw$ is a cycle consisting of both edges uv and vw in G, contradicting the statement of Theorem 2.4.3. Therefore, all xy-paths should pass through v. This completes the proof. $\square$

2.5 Cut-Sets of a Graph

Recall that an edge e of a graph G is a cut-edge of G if $G-e$ is disconnected.

Definition 2.5.1 (Edge-Cut). A subset F of the edge set E of a graph G is said to be an *edge-cut* of G if $G-E$ is disconnected.

The following graph illustrates a vertex-cut of a graph.

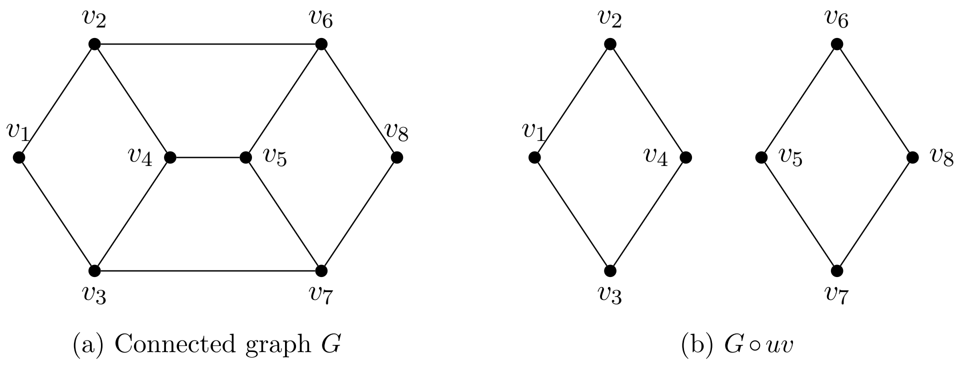


Figure 2.8: Disconnected graph $G - \{v_4v_5, v_2v_6, v_3v_7\}$.

In the above graph G, the edge set $F = \{v_4v_5, v_2v_6, v_3v_7\}$ is an edge-cut, since the removal of F makes G disconnected.

Definition 2.5.2 (Cut-Set). A *cut-set* of a *bond* of a graph G is a minimal non-empty edge-cut of G. That is, a cut-set of a graph G is a set of edges F of G whose removal makes the graph disconnected, provided the removal of no proper subset of F makes G disconnected.

A cut-set also called a *minimal cut-set*, a *proper cut-set*, a *simple cut-set*, or a *cocycle*. Note that the edge-cut F in the above example is a cut-set of G.

If a cut-set puts two vertices v_1 and v_2 into two different components, then, it is called a cut-set with regard to v_1 and v_2 .

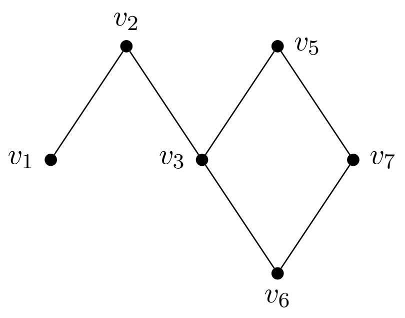
Remark 2.5.3. If e is a cut-edge of G, then $\{e\}$ is an edge-cut of G. Any set of cut-edges in G is also an edge-cut of G.

2.6 Connectivity in Graphs

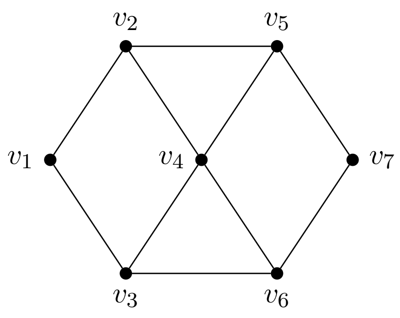
Definition 2.6.1 (Separable Graph)**.** A connected graph G (or a connected component of a graph) is said to be a *separable graph* if it has a *cut-vertex.*

Definition 2.6.2 (Non-separable Graph)**.** A connected graph or a connected component of a graph, which is not separable, is called *non-separable graph.*

Figure 2.9(a) depicts a separable graph, while Figure 2.9(b) depicts a non-separable graph.



(a) A separable graph

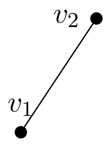


(b) A non-separable graph

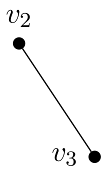
Figure 2.9: Illustration to separable and non-separable graphs

Definition 2.6.3 (Block)**.** A non-separable subgraph of a separable graph G is called a *block* of G.

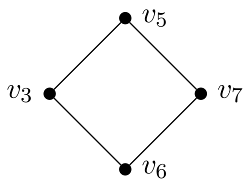
In Figure 2.9(a), v_2 and v_3 are cut-vertices. Hence, the three blocks of that graph are given in Figure 2.10:



(a) Block 1



(b) Block 2



(c) Block 3

Figure 2.10: Illustration to blocks of a graph

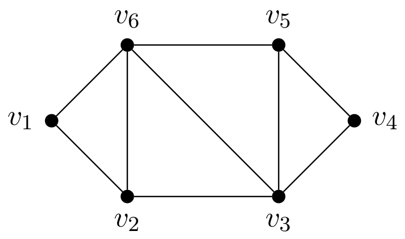
Definition 2.6.4 (Edge Connectivity of a Graph)**.** Let G be a graph (may be disconnected) having k components. The minimum number of edges whose deletion from G increases the number of components of G is called the *edge connectivity* of G. The edge connectivity of G is denoted by $\lambda(G)$.

Remark 2.6.5. Note that the number of edges in the smallest cut-set of a graph is its edge connectivity.

Theorem 2.6.6. *The edge connectivity of a graph G is less than or equal to its minimum degree. That is, $\lambda(G) \leq \delta(G)$.*

Proof. Let v_i be a vertex in G such that $d(v_i) = \delta(G)$. Then, note that v_i can be separated from G only after the removal of $d(v_i)$ edges incident on v_i. Therefore, $\lambda(G) \leq \delta(G)$. □

Problem 2.6.7. List out all cut-sets and hence determine the edge connectivity of the following graph:



Solution: The cut-sets of the given graph G are listed below:

$$\begin{aligned}
S_1 &= \{v_1v_2, v_1v_6\}\\
S_2 &= \{v_3v_4, v_4v_5\}\\
S_3 &= \{v_3v_5, v_4v_5, v_5v_6\}\\
S_4 &= \{v_2v_3, v_3v_6, v_5v_6\}\\
S_5 &= \{v_3v_4, v_3v_5, v_5v_6\}\\
S_6 &= \{v_1v_6, v_2v_3, v_2v_6\}\\
S_7 &= \{v_2v_3, v_3v_5, v_4v_5, v_3v_6\}\\
S_8 &= \{v_1v_6, v_2v_6, v_3v_6, v_5v_6\}\\
S_9 &= \{v_2v_3, v_3v_4, v_3v_5, v_3v_6\}\\
S_{10} &= \{v_1v_6, v_2v_6, v_3v_4, v_3v_5, v_3v_6\}\\
S_{11} &= \{v_1v_2, v_2v_6, v_3v_5, v_3v_6, v_4v_5\}\\
S_{12} &= \{v_1v_2, v_2v_6, v_3v_4, v_3v_5, v_3v_6\}\\
S_{13} &= \{v_1v_6, v_2v_6, v_3v_4, v_3v_5, v_4v_5\}
\end{aligned}$$

From the above list, we note that the cardinality of a minimal cut-set is 2. Therefore, edge connectivity of the given graph is 2. □

Definition 2.6.8 (Vertex Connectivity of a Graph)**.** Let G be a graph (may be disconnected). The minimum number of vertices whose deletion from G increase the number of components of G is called the *vertex connectivity* of G. The vertex connectivity of G is denoted by $\kappa(G)$.

Remark 2.6.9. Invoking Definition 2.6.8, note that every separable graph is a 1-connected graph.

Theorem 2.6.10. *The vertex connectivity of any graph G is less than or equal to the edge connectivity of G. That is, $\kappa(G) \leq \lambda(G)$.*

Proof. Let $\lambda = \lambda(G)$ and $\kappa = \kappa(G)$. Since λ is the edge connectivity of G, there exists a cut-set S in G with λ edges. Note that the maximum number of vertices to be removed from G to delete λ edges from G is λ. Therefore, $\kappa \leq \lambda$, completing the proof. □

Theorem 2.6.11. *Every cut-set in a non-separable graph with more than two vertices contains at least two edges.*

Proof. A graph is non-separable if its vertex connectivity is at least two. In view of Theorem 2.6.10, edge connectivity of a non-separable graph must be at least two which is possible if the graph has at least two edges. □

Theorem 2.6.12. *The maximum vertex connectivity of a connected graph G with n vertices and ϵ edges is $\lfloor \frac{2\epsilon}{n} \rfloor$.*

Proof. We know that $\frac{2\epsilon}{n}$ is the average degree of G (see Theorem 1.2.7) and hence there must be at least one vertex in G whose degree is less than or equal to the number $\lfloor \frac{2\epsilon}{n} \rfloor$. Also, we have $\delta(G) \leq \frac{2\epsilon}{n}$ (see Theorem 1.2.7). By Theorem 2.6.10, we have $\kappa(G) \leq \lambda(G) \leq \lfloor \frac{2\epsilon}{n} \rfloor$. Hence, $\kappa(G) \leq \lfloor \frac{2\epsilon}{n} \rfloor$. □

Theorem 2.6.13 (Whitney's Inequality)**.** *For any graph G, we have $\kappa(G) \leq \lambda(G) \leq \delta(G)$.*

Proof. The proof follows immediately from Theorem 2.6.6 and Theorem 2.6.10. □

Theorem 2.6.14. *If G is a cubic graph (3-regular graph), then $\kappa(G) = \lambda(G)$.*

Proof. Let S be a minimum vertex-cut of a cubic graph G. Then, $|S| = \kappa(G) \leq \lambda(G)$. Let H_1 and H_2 be two components of $G - S$. Since S is a minimal vertex-cut of G, every vertex in S must have a neighbour in H_1 and a neighbour in H_2. Since G is 3-regular, v cannot have two neighbours in H_1 and two neighbours in H_2.

Now, delete the edge from v to a member of $\{H_1, H_2\}$, when v has only one neighbour. If a path enters S via one vertex v_i in S and leaves for H_2 via another vertex v_j in S, adjacent to v_i, then we delete the edges to H_1 for both v_i and v_j. Clearly, the number of edges deleted here (both cases together) is equal to the number of vertices in S and these $\kappa(G)$ edges breaks all paths from H_1 to H_2. Hence, $\kappa(G) = \lambda(G)$. □

Definition 2.6.15. A graph G is said to be *k-connected* if its vertex connectivity is k (That is, if $\kappa(G) = k$). A graph G is *k-edge connected* if its edge connectivity is k (That is, if $\lambda(G) = k$.)

Theorem 2.6.16. *A graph G is k-connected if and only if there exist at least k disjoint paths (paths without common vertices) between any pair of vertices in G.*

Proof. Assume that G is k-connected. Then, k vertices are to be removed to make G disconnected. If there are fewer than k disjoint paths between any two vertices u and v in G, then u and v will be in different components on removing fewer than k vertices from G, a contradiction to the fact that G is k-connected. Therefore, there must be at least k disjoint paths between any pair of vertices in G.

Conversely, assume that there exist at least k disjoint paths between any pair of vertices u and v in G. Then, we note that the removal of k vertices, one from each one of the k disjoint paths, leaves u and v in two different components. Moreover, removing fewer than k vertices will not leave u and v in two different components. Hence, G is k-connected. □

Theorem 2.6.17. *A graph G is k-edge connected if and only if there exist at least k edge-disjoint paths (paths that may have common vertices, but no common edges) between any pair of vertices in G and at least one pair of vertices is joined by exactly k edge disjoint paths.*

Proof. The result is straight-forward from the fact that if two vertices u and v of a graph are connected by k-edge disjoint paths, then u and v are disconnected if and only if k edges, one from each of the k paths between them, are removed. □

Theorem 2.6.18 (Whitney's Theorem). *A graph with at least three vertices is a 2-connected graph if and only if there exist internally disjoint paths between any pair of vertices in G.*

Proof. The proof is a specific case of Theorem 2.6.16, where $k = 2$. □

Theorem 2.6.19. *Every cycle (circuit) in G has even number of edges in common with any cut-set of G.*

Proof. Let S be a cut-set in G. Then, $G - S$ is disconnected. Let V_1 and V_2 be the two disjoint subsets of $V(G - S)$. Any cycle which lies entirely in V_1 (or V_2) does not have any common edge with a cut-set of G. If a cycle C has vertices in both V_1 and V_2, we have to traverse back and forth between V_1 and V_2 to traverse the cycle C. Since every edge in S has one end in V_1 and the other end in V_2, the number of edges common to C and S must be even. □

Theorem 2.6.20. *Let $k \geq 2$. Show that every k-connected graph of order at least $2k$ contains a cycle of length at least $2k$.*

Proof. Let C be a maximal cycle in G. If G is a cycle, then the proof is complete. Otherwise, we will have at least one vertex, say$v \in V(G)$, such that $v \notin C$. Since G is k-connected, as a consequence of Theorem 2.6.16, there exists k disjoint paths, say $P_1, P_2, \ldots, P_k$ from v to C. If $|C| < 2k$, then there exists $i, j \in \{1, 2, \ldots, k\}$, such that P_i and P_j are connected to adjacent vertices in C, which allows to construct a longer cycle than C, a contradiction to the maximality of C. Therefore, $|C| = 2k$, completing the proof. □

The above theorem can easily be verified from the graph G in Figure 2.8.

Theorem 2.6.21. *The ringsum of any two cut-sets in a connected graph G is a cut-set or an edge-disjoint union of cut-sets.*

Proof. Let S_1 and S_2 be two cut-sets in a connected graph G. Let (V_1, V_2) be a unique partition of $V(G)$ with respect to S_1, whereas (U_1, U_2) be a unique partition of $V(G)$ with respect to S_2. Then, we have

$$V_1 \cup V_2 = V; V_1 \cap V_2 = \emptyset;$$
$$V_3 \cup V_4 = V; V_3 \cap V_4 = \emptyset.$$

Now, we have

$$(V_1 \cap V_4) \cup (V_2 \cap V_3) = V_1 \oplus V_3 = V_5;$$
$$(V_1 \cap V_3) \cup (V_2 \cap V_4) = V_2 \oplus V_4 = V_6.$$

It is to be noted that the ringsum $S_1 \oplus S_2$ consists of the edges that join the vertices in V_5 to those in V_6. Thus, the set of edges $S_1 \oplus S_2$ partitions $V(G)$ into two sets V_5 and V_6 such that

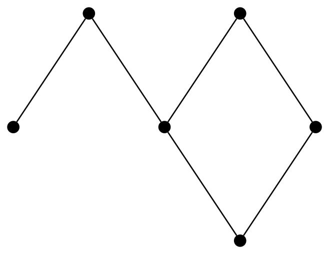
$$V_1 \cup V_2 = V \; ; \; V_1 \cap V_2 = \emptyset.$$

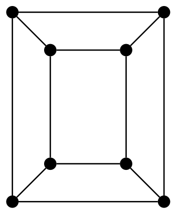
Hence, $S_1 \oplus S_2$ is a cut-set if subgraphs containing V_5 and V_6 remain connected in $G - (S_1 \oplus S_2)$. Otherwise, $S_1 \oplus S_2$ is just an edge-disjoint union of two cut-sets. □

2.7 Exercise Problems

1. Show that every uv-walk contains a uv-path.
2. Show that every closed walk contains a cycle.
3. Show that every graph with n vertices and k edges, $n > k$ has $n - k$ components.
4. If every vertex of a graph G has degree greater than or equal to 2, then G has some cycles.
5. If G is a simple graph with $d(v) \geq k$, $\forall v \in V(G)$, then G contains a path of length at least k. If $k \geq 2$, then G contains a cycle of length $k+1$.
6. Show that if G is simple and $\delta(G) \geq k$, then G has a path of length k.
7. If a connected graph G is decomposed into two subgraphs G_1 and G_2, then show that there must be at least one common vertex between G_1 and G_2.
8. If we remove an edge e from a graph G and $G - e$ is still connected, then show that e lies along some cycle of G.
9. If the intersection of two paths is a disconnected graph, then show that the union of the two paths has at least one cycle.

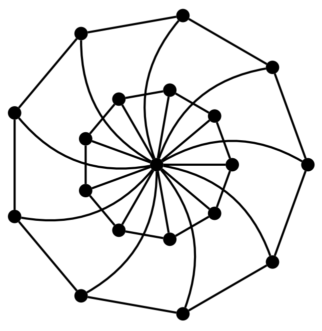
10. If P_1 and P_2 are two different paths between two given vertices of a graph G, then show that $P_1 \oplus P_2$ is a cycle or a set of cycles in G.
11. Show that the complement of a complete bipartite graph is the disjoint union of two complete graphs.
12. For a simple graph G, with n vertices, if $\delta(G) = \frac{n-1}{2}$, then G is connected.
13. Show that any two longest paths in a connected graph have a vertex in common.
14. For $k \geq 2$, prove that a k-regular bipartite graph has no cut-edge.
15. Determine the maximum number of edges in a bipartite subgraph of the Petersen graph.
16. If H is a subgraph of G, then show that $d_G(u,v) \leq d_H(u,v)$.
17. Prove that if a connected graph G has equal order and size, then G is a cycle.
18. Show that eccentricities of adjacent vertices differ by at most 1.
19. Prove that if a graph has more edges than vertices then it must possess at least one cycle.
20. If the intersection of two paths is a disconnected graph, then show that the union of the two paths has at least one cycle.
21. Find the blocks and connectivity of the following graphs:

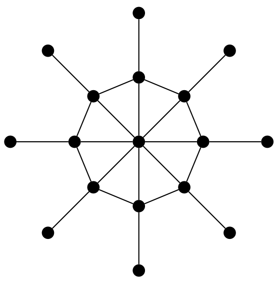


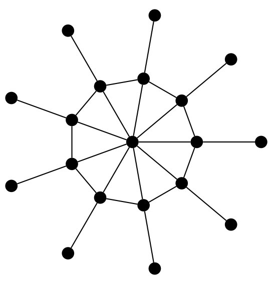


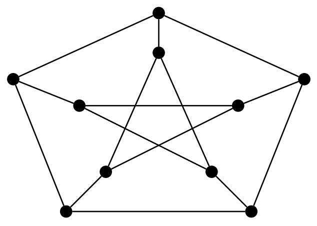
22. Show that every edge-cut is a disjoint union of bonds.
23. If G has a cut-edge, then show that G has a vertex v such that $\omega(G-v) > \omega(G)$. Is the converse true? Justify your answer.
24. Show that a simple connected graph that has exactly two vertices which are not cut-vertices is a path.
25. Show that if G is simple and $\delta > v-2$, then $\kappa(G) = 8$.
26. Show that if G has no even cycles, then each block of G is either K_1 or K_2, or an odd cycle.
27. Show that a connected graph which is not a block has at least two blocks that each contains exactly one cut vertex.
28. Let G be a connected graph with at least three vertices. Prove that if G has bridge then G has a cut vertex.
29. Let u and v be two vertices of a 2-connected graph. Then, show that there is a cycle in G passing through both u and v.
30. Show that a cut-vertex of a graph is not a cut-vertex of its complement.
31. Show that $\kappa(G) \leq \frac{2\epsilon}{n}$.
32. The ring sum of two distinct proper edge-cut-sets is an edge-cut-set.

33. Prove that any two connected graphs with n vertices, all of degree 2, are isomorphic.
34. Find the eccentricity of the vertices and the radius, the diameter and center(s) of the following graphs:









****

Chapter 3

Traversability in Graphs

****

3.1 Königsberg Seven Bridge Problem

The Seven Bridges of Königsberg is a historically notable problem in Mathematics and was instrumental in the origination of Graph theory as a branch of modern Mathematics.

The city of *Königsberg* in Prussia (the capital of East Prussia at that time) (now Kaliningrad, Russia) was situated on either sides of the *Pregel River* and included two large islands which were connected to each other and the mainlands by *seven bridges* (see Figure 3.1).

The problem was to devise a walk through the city that would cross each bridge once and only once, subject to the following conditions:

(i) The islands could only be reached by the bridges;
(ii) Every bridge once accessed must be crossed to its other end;
(iii) The starting and ending points of the walk are the same.

In 1736, a Swiss Mathematician *Leonard Euler* introduced a graphical model to this problem by representing each land area by a vertex and each bridge by an edge connecting corresponding vertices (see the following figure).

Using this graphical model, Euler proved that no such walk (trail) exists.

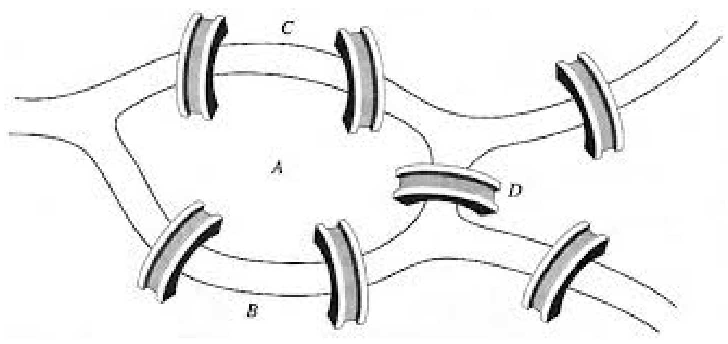


Figure 3.1: Königsberg's seven bridge problem.

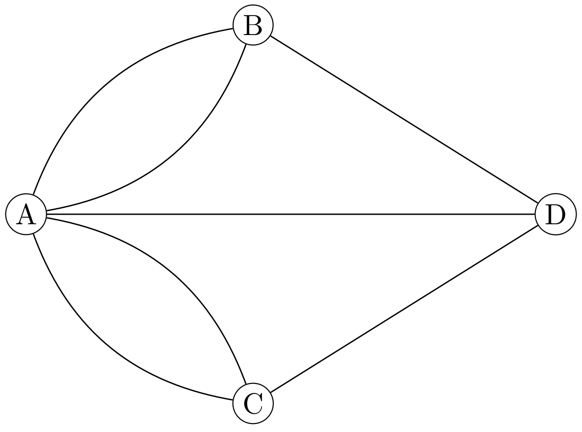


Figure 3.2: Graphical representation of seven bridge problem

Two of these seven original bridges were destroyed in the bombing of Königsberg in the second world war. Two others were later demolished for the construction of a new highway and one bridge was rebuilt in 1935. The remaining two bridges still remain exactly as they were in Euler's time.

The Königsberg seven bridge problem is the same as the problem of drawing figures in a paper without lifting the pen from the paper.

3.2 Eulerian Graphs

A *tour* of a graph G is a closed walk that traverses each edge of G at least once.

Definition 3.2.1 (Traversable Graph)**.** An *Eulerian trail* or *Euler walk* in an undirected graph is a walk that uses each edge exactly once. If an Euler trail exists in a given graph G, then G is called a *traversable graph* or a *semi-Eulerian graph.*

Definition 3.2.2 (Eulerian Graph)**.** An *Euler tour* in an undirected graph is a cycle that uses each edge exactly once. If such an Euler cycle exists in the graph concerned, then the graph is called an *Eulerian graph* or a *unicursal graph.*

The following theorem characterises the class of Eulerian graphs:

Theorem 3.2.3 (Euler Theorem)**.** *A connected graph G is Eulerian if and only if every vertex in G is of even degree.*

Proof. If G is Eulerian, then there is an Euler tour, say P, in G. Every time a vertex is listed, that accounts for two edges adjacent to that vertex, the one before it in the list and the one after it in the list. This closed walk uses every edge exactly once. So, every edge is accounted for and there are no repeats. Thus every degree must be even.

Conversely, let us assume that each vertex of G has even degree. We need to show that G is Eulerian. We prove the result by induction on the number of edges of G. Let us start with a vertex $v_0 \in V(G)$. As G is connected, there exists a vertex $v_1 \in V(G)$ that is adjacent to v_0. Since G is a simple graph and $d(v_i) \geq 2$, for each vertex $v_i \in V(G)$, there exists a vertex $v2 \in V(G)$, that is adjacent to v_1 with $v_2 \neq v_0$. Similarly, there exists a vertex $v_3 \in V(G)$, that is adjacent to v_2 with $v_3 \neq v_1$. Note that either $v_3 = v_0$, in which case, we have a cycle $v_0v_1v_2v_0$ or else one can proceed as above to get a vertex $v_4 \in V(G)$ and so on. As the number of vertices is finite, the process of getting a new vertex will finally end with a vertex v_i being adjacent to a vertex v_k, for some i, $0 \leq i \leq k-2$. Hence, $v_i - v_{i+1} - v_{i+2} - \ldots - v_k - v_i$ forms a cycle, say C, in G.

If C contains every edge of G, then C gives rise to a closed Eulerian trail and we are done. So, let us assume that $E(C)$ is a proper subset of $E(G)$. Now, consider the graph G_1 that is obtained by removing all the edges in C from G. Then, G_1 may be a disconnected graph but each vertex of G_1 still has even degree. Hence, we can do the same process explained above to G_1 also to get a closed Eulerian trail, say C_1. As each component of G_1 has at least one vertex in common with C, if C_1 contains all edges of G_1, then $C \cup C_1$ is a closed Euler trail in G. If not, let G_2 be the graph obtained by removing the edges of C_1 from G_1. (That is, $G_2 = G_1 - E(C \cup C_1)$).

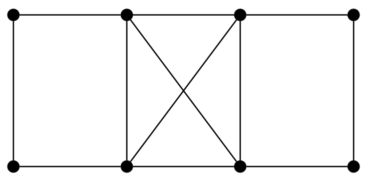
Since G is a finite graph, we can proceed to find out a finite number of cycles only. Let the process of finding cycles, as explained above, ends after a finite number of steps, say r. Then, the reduced graph $G_r = G_{r-1} - E(C_{r-1}) = G - E(C \cup C_1 \cup C_{r-1})$ will be an empty graph (null graph). Then, $C \cup C_1 \cup C_2 \ldots \cup C_{r-1}$ is a closed Euler trail in G. Therefore, G is Eulerian. This completes the proof. □

Illustrations to an Eulerian graph and a non-Eulerian graph are given in Figure 3.3.

In the first graph in Figure 3.3, every vertex has even degree and hence by Theorem 3.2.3, it is Eulerian. In the second graph, some vertices have odd degree and hence it is not Eulerian.

Note: In an Euler graph, it can be noted that every edge of G is contained in exactly one cycle of G. Hence, we have the following Theorem.

Problem 3.2.4. Is the following graph G Eulerian? If yes, write an Euler tour (Euler line) from G.


(a) An Eulerian graph.

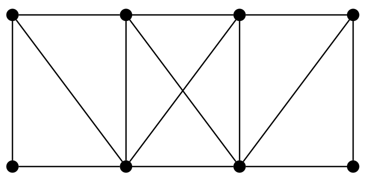

(b) A non-Eulerian graph

Figure 3.3: Examples of Eulerian and non-Eulerian graphs

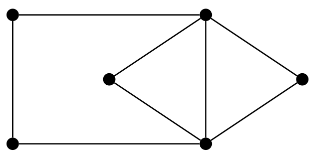


Figure 3.4: A graph G to check whether Eulerian or not.

Solution: Label the vertices of G as shown in Figure 3.5 given below:

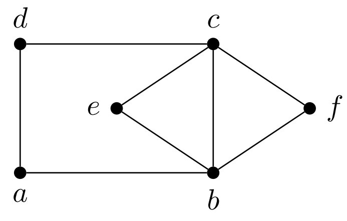


Figure 3.5: An Eulerian graph G.

Clearly, all vertices in G are even degree vertices. Then, by Theorem 3.1, G is an Eulerian graph. We can find an Euler tour in G starting from a as : $a - b - e - c - b - f - c - d - a$. □

Theorem 3.2.5. *A connected graph G is Eulerian if and only if it can be decomposed into edge-disjoint cycles.*

Proof. Assume that G can be decomposed into edge-disjoint cycles. Since the degree of every vertex in a cycle is 2, the degree of every vertex in G is two or multiples of 2. That is, all vertices in G are even degree vertices. Then, by Theorem 3.2.3, G is Eulerian.

Converse part is exactly the same as that of Theorem 3.2.3. □

Theorem 3.2.6. *A connected graph G is traversable if and only if it has exactly two odd degree vertices.*

Proof. In a traversable graph, there must be an Euler trail. The starting vertex and terminal vertex need not be the same. Therefore, these two vertices can have odd degrees. Remaining part of the theorem is exactly as in the proof of Theorem 3.2.3. □

Theorem 3.2.7. *An Eulerian graph does not have a bridge.*

Proof. Let G be an Eulerian graph. Then, by Theorem 3.2.5, G can be decomposed into cycles. That is, every edge of G is a part of (exactly) one cycle of G. Then, by Theorem 2.3.5, G has no cut-edge, completing the proof. □

Theorem 3.2.8. *An Eulerian graph cannot have a cut-set of an odd number of edges.*

Proof. Let G be an Eulerian graph. Then, it can be decomposed into edge-disjoint cycles. First, suppose that two cycles among them have exactly one common vertex. Then, we have to remove two edges of any one cycle to make the graph disconnected. Next, assume that G is obtained by k cycles with two common vertices. Then, to make G disconnected, we need to remove two edges from each of these cycles. Therefore, the number edges in any cut-sets of an Eulerian graph is even. □

3.3 Fleury's Algorithm

Fleury's algorithm describes a procedure which constructs an Euler tour in an Eulerian graph. The algorithm can be described as goven below:

Fleury's Algorithm:

(i) Choose any vertex v_1 in the Euler graph G and set $W_0 = \{v_0\}$.

(ii) If the trail $W_i = v_1e_1v_2e_2v_3 \dots e_{i-1}v_i$ has been chosen, such that all edges selected are all different, then choose another edge e_i such that

 (a) e_i is different from all previously selected edges $e_1, e_2, \dots, e_i$;

 (b) e_i is incident with v_i; and

 (c) unless there is no alternative, e_i is not a cut-edge of the edge deleted subgraph $G - \{e_1, e_2, \dots, e_{i-1}\}$.

(ii) If W_i covers every edge of G, stop. Otherwise, go to step (ii).

Step by step Illustration

Consider the following graph in Figure 3.6. Since every vertex of G has even degree, G is an Euler graph.

A step by step illustration of Flery's Algorithm is given below. White vertices and dashed lines denote the vertices and edges selected at different steps.

Step-1: Choose $v_1 = u_1$ and let $W_1 = \{v_1\}$. Choose the edge $e_1 = u_1u_2$. Then, $W = \{u_1e_1u_2\}$ and we have

Step 2: Choose the edge $e_2 = u_2u_6$. Then, $W = \{u_1e_1u_2e_2u_6\}$ and we have

Step 3: Choose the edge $e_3 = u_6u_8$. Then, $W = \{u_1e_1u_2e_2u_6e_3u_8\}$ and we have

Step 4: Choose the edge $e_4 = u_8u_7$. Then, $W = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7\}$ and we have

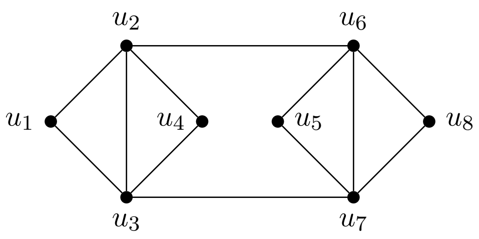
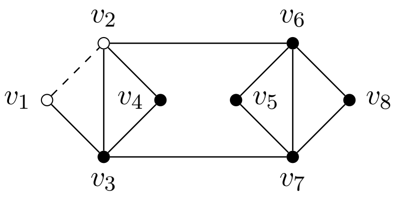
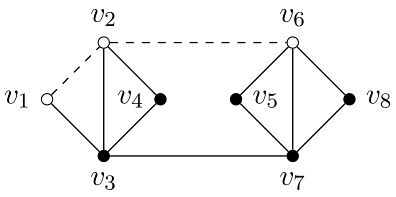
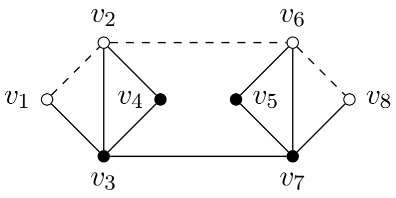
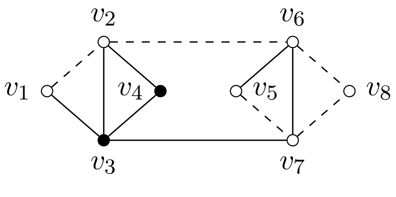


Figure 3.6: An Eulerian graph

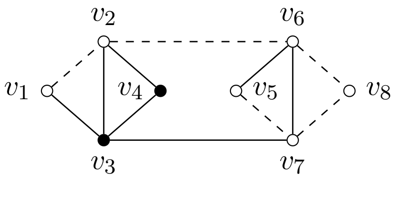




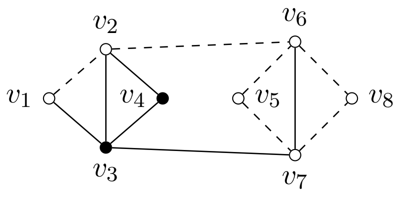




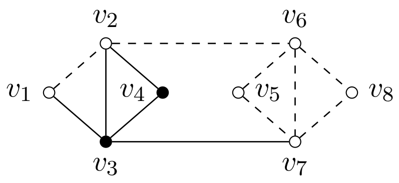
Step 5: Choose the edge $e_5 = u_7u_5$. Then, $W = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7e_5u_5\}$ and we have



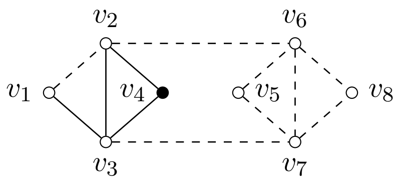
Step 6: Choose the edge $e_6 = u_5u_6$. Then, $W = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7e_5u_5e_6u_6\}$ and we have



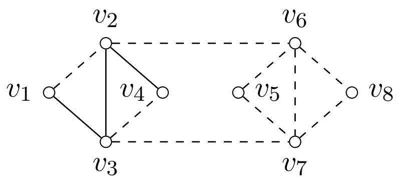
Step 7: Choose the edge $e_7 = u_6u_7$. Then, $W = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7e_5u_5e_6u_6$ $e_7u_7\}$ and we have



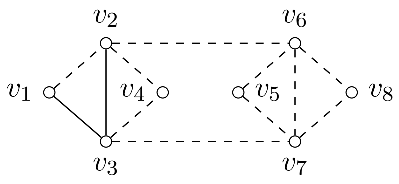
Step 8: Choose the edge $e_8 = u_7u_3$. Then, $W = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7e_5u_5e_6u_6$ $e_7u_7e_8u_3\}$ and we have



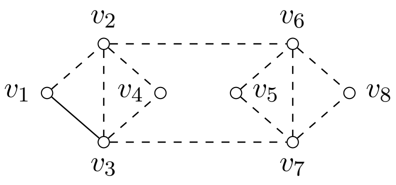
Step 9: Choose the edge $e_9 = u_3u_4$. Then, $W_4 = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7e_5u_5e_6u_6$ $e_7u_7e_8u_3e_9u_4\}$ and we have



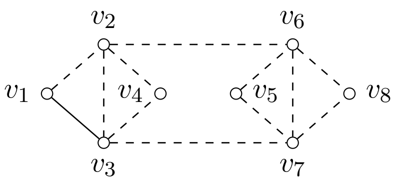
Step 10: Choose the edge $e_{10} = u_4u_2$. Then, $W_4 = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7e_5u_5e_6u_6$ $e_7u_7e_8u_3e_9u_4e_{10}u_2\}$ and we have



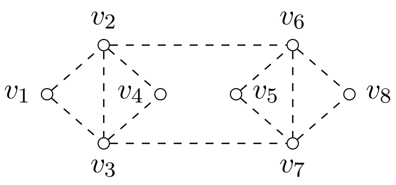
Step 11: Choose the edge $e_1 = u_2u_3$. Then, $W_4 = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7e_5u_5e_6u_6$ $e_7u_7e_8u_3e_9u_4e_{10}u_2e_{11}u_3\}$ and we have



Step 11: Choose the edge $e_{11} = u_2u_3$. Then, $W_4 = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7e_5u_5e_6u_6$ $e_7u_7e_8u_3e_9u_4e_{10}u_2e_{11}u_3\}$ and we have



Step 12: Choose the edge $e_{12} = u_3u_1$. Then, $W_4 = \{u_1e_1u_2e_2u_6e_3u_8e_4u_7e_5u_5e_6u_6$ $e_7u_7e_8u_3e_9u_4e_{10}u_2e_{11}u_3e_{12}u_1\}$ and we have



Hence, the walk W covers all edges in G and hence it is an Eulerian tour in G.

3.4 Chinese Postman Problem

In his job, a postman picks up mail at the post office, delivers it, and then returns to the post office. He must, of course, cover each street in his area at least once. Subject to this condition, he wishes to choose his route in such a way that walks as little as possible. This problem is known as the Chinese postman problem, since it was first considered by a Chinese mathematician, Guan in 1960.

We refer to the street system as a weighted graph (G, w) whose vertices represent the intersections of the streets, whose edges represent the streets (one-way or two-way) and the weight represents the distance between two intersections, of course, a positive real number. A closed walk that covers each edge at least once in G is called a *postman tour*. Clearly, the Chinese postman problem is just that of finding a minimum-weight postman tour. We will refer to such a postman tour as an optimal tour.

An algorithm for finding an optimal Chinese postman route is as follows:

S-1 : List all odd vertices.
S-2 : List all possible pairings of odd vertices.
S-3 : For each pairing find the edges that connect the vertices with the minimum weight.
S-4 : Find the pairings such that the sum of the weights is minimised.
S-5 : On the original graph add the edges that have been found in Step 4.
S-6 : The length of an optimal Chinese postman route is the sum of all the edges added to the total found in Step 4.
S-7 : A route corresponding to this minimum weight can then be easily found.

Example 3.4.1. Consider the following weighted graph:

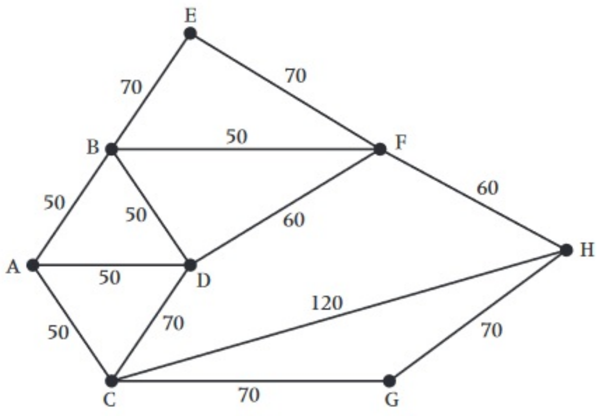


Figure 3.7: An example for Chinese Postman Problem

1. The odd vertices are A and H; There is only one way of pairing these odd vertices, namely AH;
2. The shortest way of joining A to H is using the path $\{AB, BF, FH\}$, a total length of 160;
3. Draw these edges onto the original network.
4. The length of the optimal Chinese postman route is the sum of all the edges in the original network, which is $840m$, plus the answer found in Step 4, which is $160m$. Hence the length of the optimal Chinese postman route is $1000m$.
5. One possible route corresponding to this length is $ADCGHCABDFBEFHFBA$, but many other possible routes of the same minimum length can be found.

3.5 Hamiltonian Graphs

Definition 3.5.1 (Traceable Graphs)**.** A *Hamiltonian path* (or *traceable path*) is a path in an undirected (or directed graph) that visits each vertex exactly once. A graph that contains a Hamiltonian path is called a *traceable graph.*

Definition 3.5.2 (Hamiltonian Graphs)**.** A *Hamiltonian cycle*, or a *Hamiltonian cycle*, or a *vertex tour* or a *graph cycle* is a cycle that visits each vertex exactly once (except for the vertex that is both the start and end, which is visited twice). A graph that contains a Hamiltonian cycle is called a *Hamiltonian graph.*

In Figure 3.8, the cycle $v_1 - v_2 - v_6 - v_7 - v_3 - v_4 - v_8 - v_1$ is a Hamilton cycle in G, where as $v_1 - v_2 - v_6 - v_7 - v_3 - v_4 - v_8$ is a Hamilton path in G.

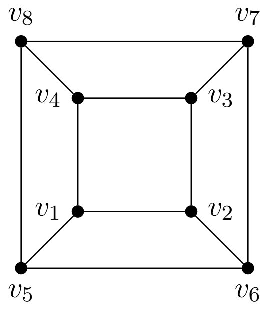


Figure 3.8: A Hamiltonian graph

Note that a graph can have a Hamilton path, but not a Hamilton cycle. The graph in Figure 3.9 is a non-Hamiltonian graph which consists of a Hamilton path. In this graph, $b - a - d - c$ is a Hamilton path, but the graph does not have a Hamilton cycle (and hence is non-Hamiltonian).

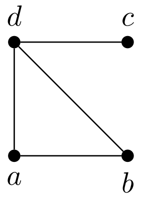


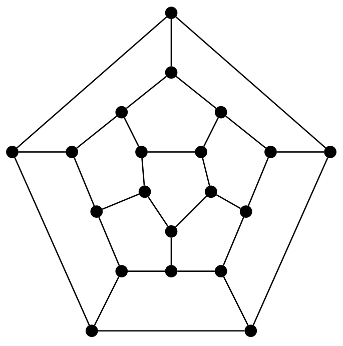
Figure 3.9: A traceable graph which is not Hamiltonian

Hamiltonian graphs are named after the famous mathematician *William Rowan Hamilton* who invented the Hamilton's puzzle, which involves finding a Hamiltonian cycle in the edge graph of the dodecahedron.

A necessary and sufficient condition for a graph to be a Hamiltonian is still to be determined. But there are a few sufficient conditions for certain graphs to be Hamiltonian. The following theorem is one of those results.

Theorem 3.5.3 (Dirac's Theorem)**.** *Every graph G with $n \geq 3$ vertices and minimum degree $\delta(G) \geq \frac{n}{2}$ has a Hamilton cycle.*

Proof. Suppose that $G = (V, E)$ satisfies the hypotheses of the theorem. Then G is connected, since otherwise the degree of any vertex in a smallest component C of G would be at most $|C| - 1 < \frac{n}{2}$, contradicting the hypothesis $\delta(G) \geq \frac{n}{2}$.


(a)

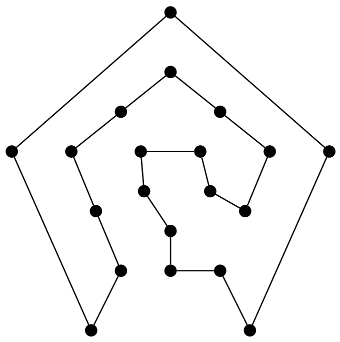
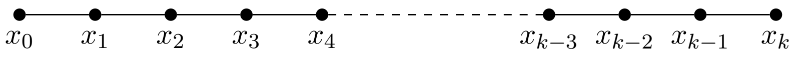
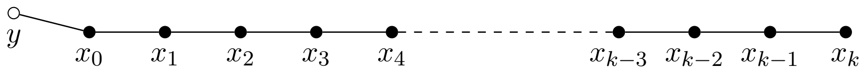

(b)

Figure 3.10: Dodecahedron and a Hamilton cycle in it.

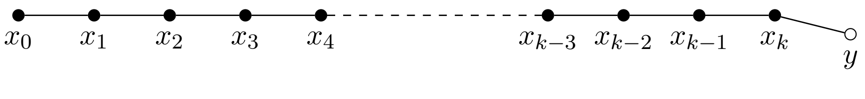


Let $P = x_0x_1 \ldots x_k$ be a longest path in G, as seen in the figure given below:

Note that the length of P is k. Since P cannot be extended to a longer path, all the neighbours of x_0 lie on P. Assume the contrary. Let y be an adjacent vertex of x_0 which is not in P. Then, the path $P' = yx_0, x_2 \ldots x_k$ is a path of length $k+1$ (see the graph given below), contradicting the hypothesis that P is the longest path in G.



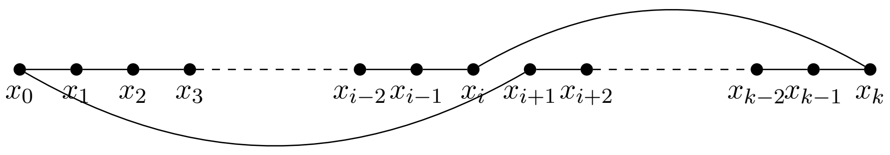
Similarly, we note that and all the neighbours of x_k will also lie on P, unless we reach at a contradiction as mentioned above (see the graph given below).

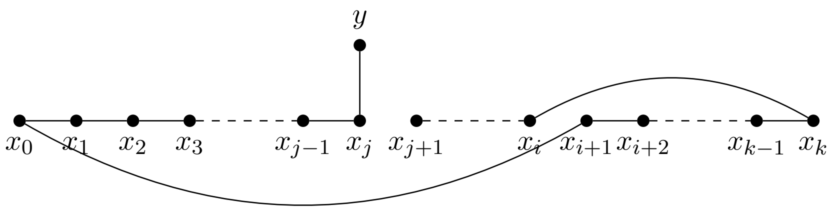


Hence, at least $\frac{n}{2}$ of the vertices $x_0, \ldots, x_{k-1}$ are adjacent to x_k, and at least $\frac{n}{2}$ of the vertices $x_1, \ldots, x_k$ are adjacent to x_0. Another way of saying the second part of the last sentence is: at least $\frac{n}{2}$ of the vertices $x_i \in \{x_0, \ldots, x_{k-1}\}$ are such that $x_0x_{i+1} \in E$. Combining both statements and using the pigeon-hole principle, we see that there is some x_i with $0 \le i \le k-1$, $x_ix_k \in E$ and $x_0x_{i+1} \in E$.

Consider the cycle $C = x_0x_{i+1}x_{i+2} \ldots x_{k-1}x_kx_ix_{i-1} \ldots x_1x_0$ as given in the following graph.

We claim that the above cycle C is a Hamilton cycle of G. Otherwise, since G is connected, there would be some vertex x_j of C adjacent to a vertex y not in C, so that $e = x_jy \in E$. But then we could attach e to a path ending in x_j containing k edges of C, constructing a path in G longer than P (see the graph given below), which is a contradiction to the hypothesis that P is the longest path in G.





Therefore, C must cover all vertices of G and hence it is a Hamiltonian cycle in G. This completes the proof. □

Dirac's Theorem is not a necessary condition for a simple graph to have a Hamilton cycle as there exist many Hamilton graphs which do not satisfy the conditions $\delta(G) \geq \frac{n}{2}$. For example, the dodecahedron graph is a Hamilton graph which does not satisfy the above condition.

Problem 3.5.4. 19 students in a nursery school play a game each day, where they hold hands to form a circle. For how many days can they do this, with no students holding hands with the same playmates more than once? Substantiate your answer with graph theoretic concepts.

Solution: Consider each student as vertex. Two vertices are adjacent if the corresponding students hold their hands. Then, the problem is reduced to finding the number of edge-disjoint Hamiltonian cycles in the corresponding graph. By Dirac's theorem (Theorem 3.5.3), this number is $\frac{n-1}{2} = \frac{19-1}{2} = 9$. □

Theorem 3.5.5 (Ore's Theorem)**.** *Let G be a graph with n vertices and let u and v be non-adjacent vertices in G such that $d(u)+d(v) \geq n$. Let $G+uv$ denote the super graph of G obtained by joining u and v by an edge. Then G is Hamiltonian if and only if $G+uv$ is Hamiltonian.*

Proof. Let G be a graph with n vertices and suppose u and v are non-adjacent vertices in G such that $d(u)+d(v) \geq n$. Let $G' = G+uv$ be the super graph of G obtained by adding the edge uv. Note that, except for u and v, $d_G(x) = d_{G'}(x) \forall x \in V(G)$.

Let G be Hamiltonian. The only difference between G and G' is the edge uv. Then, obviously G' is also Hamiltonian as a Hamilton cycle in G will be a Hamilton cycle in G' as well.

Conversely, let G' be Hamiltonian. We have to show that G is Hamiltonian. Assume the contrary. Then, by (contrapositive of) Dirac's Theorem, we have

$\delta(u) < \frac{n}{2}$ and $\delta(u) < \frac{n}{2}$ and hence we have $d(u)+d(v) < n$, which contradicts the hypothesis that $d(u)+d(v) \geq n$. Hence G is Hamiltonian. □

The following theorem determines the number of edge-disjoint Hamilton cycles in a complete graph K_n, where n is odd.

Theorem 3.5.6. *In a complete graph K_n, where $n \geq 3$ is odd, there are $\frac{n-1}{2}$ edge-disjoint Hamilton cycles.*

Proof. Note that a complete graph has $\frac{n(n-1)}{2}$ edges and a Hamilton cycle in K_n contains only n edges. Therefore, the maximum number of edge-disjoint hamilton cycles is $\frac{n-1}{2}$.

Now, assume that $n \geq 3$ and is odd. Construct a subgraph G of K_n as explained below:

The vertex v_1 is placed at the centre of a circle and the remaining $n-1$ vertices are placed on the circle, at equal distances along the circle such that the angle made at the centre by two points is $\frac{360}{n-1}$ degrees. The vertices with odd suffixes are placed along the upper half of the circle and the vertices with even suffixes are placed along the lower half circle. Then, draw edges v_iv_{i+1}, where $1 \leq i \leq n$, with the meaning that $v_{n+1} = v_1$, are drawn as shown in Figure 3.11.

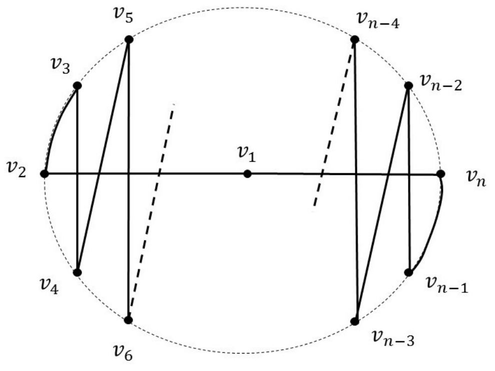


Figure 3.11: A Hamilton cycle G of K_n.

Clearly, the reduced graph G_1 is a cycle covering all vertices of K_n. That is If we rotate the vertices along the curve for $\frac{360}{n-1}$ degrees, we get another Hamilton subgraph G_2 of K_n, which has no common edges with G_1. (see figure 3.12).

In a similar way, rotate the polygonal pattern clockwise by $\frac{360}{(n-1)}$ degrees. After $(n-1)$-th rotation, all vertices will be exactly as in Figure 3.11. Therefore, $n-1$ rotations are valid. But, it can be noted that the cycle G_i obtained after the i-th rotation and the cycle $G_{\frac{n-1}{2}+i}$ are isomorphic graphs (see Figure 3.13), because all vertices in the upper half cycle in G_i will be in the lower half cycle in $G_{\frac{n-1}{2}+i}$ and vice versa, in the same order.

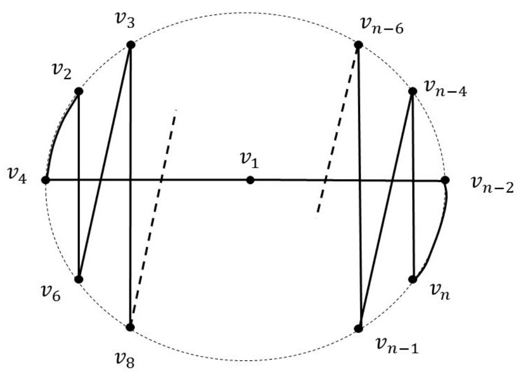
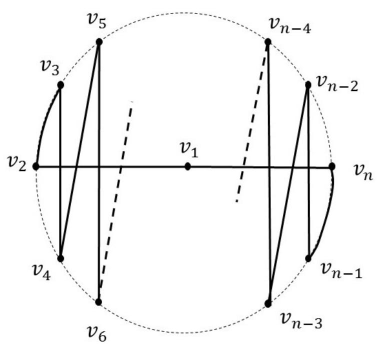


Figure 3.12: Another Hamilton cycle G_2 of K_n.



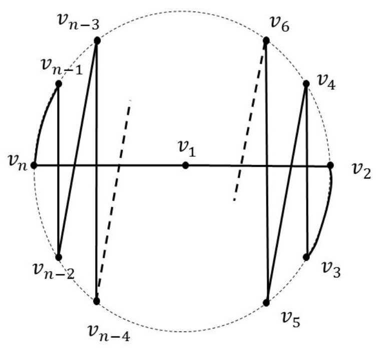


Figure 3.13: Isomorphic Hamilton cycles G_1 and $G_{\frac{n-1}{2}+1}$ of K_n.

That is, we have now that there are $\frac{n-1}{2}$ distinct such non-isomorphic edge-disjoint cycles in K_n. Hence, the number of edge-disjoint Hamilton cycles is $\frac{n-1}{2}$. □

A Hamilton cycle decomposition of a graph G is a decomposition of G into Hamilton cycles. Hence, the above theorem can be re-stated as follows:

Theorem 3.5.7. *A complete graph of odd order is Hamilton cycle decomposable. Moreover, this decomposition consists of $\frac{n-1}{2}$ edge-disjoint Hamilton cycles of the graph.*

Theorem 3.5.8. *Let G be a connected r-regular graph of even order n such that its complement is also connected. Then,*

(i) either G or its complement is Eulerian.

(ii) either G or its complement is Hamiltonian.

Proof. (i) Since G is an r-regular graph, $\overline{G}$ is $(n-r-1)$-regular graph. Given that n is even. Then, if r is even, then by Theorem 3.2.3 G is Eulerian. Moreover, $n-r-1$ will be odd and hence $\overline{G}$ is not Eulerian. Now assume that r is odd.

Therefore, G is not Eulerian. But, $n-r-1$ is even and hence by Theorem 3.2.3, $\overline{G}$ is an Eulerian graph.

(ii) If G is Hamiltonian, the proof is complete. If G is non-Hamiltonian, then by Dirac's theorem, $r < \frac{n}{2}$. If possible, assume that $\overline{G}$ is also non-Hamiltonian. Then, $n-r-1 < \frac{n}{2}$. Hence, for any vertex $v \in V(G)$, we have

$$\begin{aligned} d_G(v) &= \qquad r \le \frac{n}{2} - 1 \\ d_{\overline{G}}(v) &= n-r-1 \le \frac{n}{2} - 1. \end{aligned}$$

Hence, $d_G(v)+d_{\overline{G}}(v) \le n-2$, a contradiction to the fact that $d_G(v)+d_{\overline{G}}(v) = n-1$. Therefore, $\overline{G}$ is Hamiltonian. This complete the proof. □

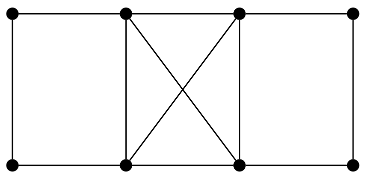
Problem 3.5.9. Every Hamilton path of a graph G is a spanning tree of G.

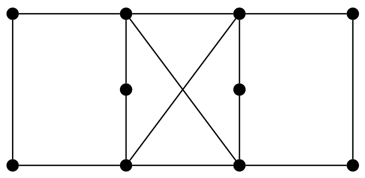
Solution: Let P be a Hamilton path in a given graph G. Then, P covers all vertices of G, making it connected and acyclic. Therefore, P is a spanning tree of G. □

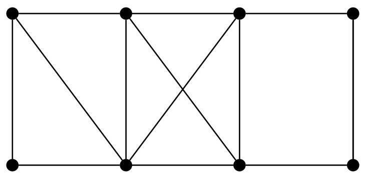
3.6 Some Illustrations

We can find out graphs, which are either Eulerian or Hamiltonian or simultaneously both, whereas some graph are neither Eulerian nor Hamiltonian. We note that the dodecahedron is an example for a Hamiltonian graph which is not Eulerian (see Figure 3.10a).

Let us now examine some examples all possible types of graphs in this category.


(a) A graph which is both Eulerian and Hamiltonian


(b) A graph which is Eulerian, but not Hamiltonian.


(c) A graph which is Hamiltonian, but not Eulerian.

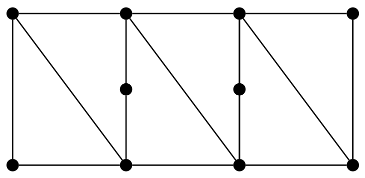

(d) A graph which is neither Eulerian nor Hamiltonian.

Figure 3.14: Traversability in graphs

3.7 Weighted Graphs

Definition 3.7.1 (Weighted Graphs)**.** A *weighted graph* is a graph G in which each edge e has been assigned a real number $w(e)$, called the *weight* (or *length*) of the edge e. The *weight of a graph G* is the sum of weights of all its edges.

Figure 3.15 illustrates a weighted graph:

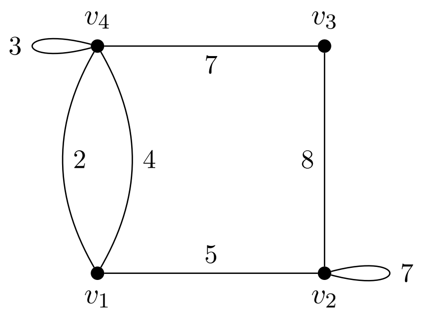


Figure 3.15: An example of a weighted graph

If H is a subgraph of a weighted graph, the weight $w(H)$ of H is the sum of the weights $w(e_1)+w(e_2)+\ldots+w(e_k)$ where $\{e_1,e_2,\ldots,e_k\}$ is the set of edges of H.

Many optimisation problems amount to finding, in a suitable weighted graph, a certain type of subgraph with minimum (or maximum) weight.

3.8 Travelling Salesman's Problem

Suppose a travelling salesman's territory includes several towns with roads connecting certain pairs of these towns. As a part of his work, he has to visit each town. For this, he needs to plan a round trip in such a way that he can visit each of the towns exactly once.

We represent the salesman's territory by a weighted graph G where the vertices correspond to the towns and two vertices are joined by a weighted edge if and only if there is a road connecting the corresponding towns which does not pass through any of the other towns, the edge's weight representing the length of the road between the towns.

Then, the problem reduces to check whether the graph G is a Hamiltonian graph and to construct a Hamiltonian cycle of minimum weight (or length) if G is Hamiltonian. This problem is known as the *Travelling Salesman Problem.*

It is sometimes difficult to determine if a graph is Hamiltonian as there is no easy characterisation of Hamiltonian graphs. Moreover, for a given a weighted graph G which is Hamiltonian there is no easy or efficient algorithm for finding an optimal ccycle in G, in general. These facts make our problem difficult.

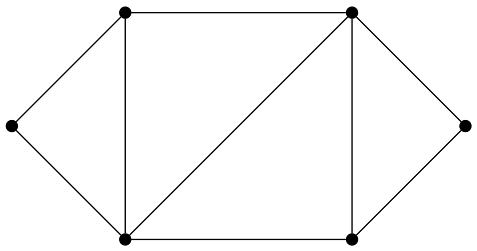
To find out an optimal Hamilton cycle, we use the following algorithm with an assumption that the given graph G is a weighted complete graph.

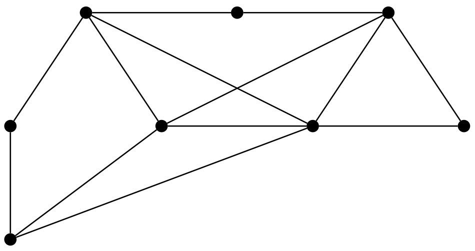
Two Optimal Algorithm

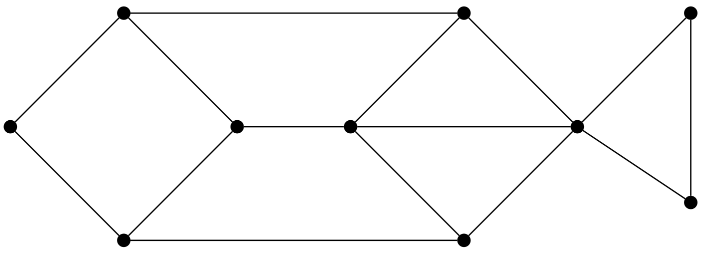
1. Let $C = v_1v_2, v_3\dots, v_nv_1$ be any Hamiltonian cycle of the weighted graph G and let w be the weight of C. That is, $w = w(v_1v_2) + w(v_2, v_3) + ... + w(v_nv_1)$.
2. Set $i = 1$.
3. Set $j = i+2$.
4. Let $C_{ij} = v_1v_2v_2v_3\dots v_iv_jv_{j-1}\dots v_{i+1}v_{j+1}v_{j+2}\dots v_nv_1$ be the Hamiltonian cycle and let w_{ij} denote the weight of C_{ij}, so that $w_{ij} = w - w(v_iv_{i+1}) - w(v_jv_{j+1}) + w(v_iv_j) + w(v_{i+1}v_{j+1})$.
   If $w_{ij} < w$, (that is, if $w(v_iv_j) + w(v_{i+1}v_{j+1}) < w(v_iv_{i+1}) + w(v_jv_{j+1})$), then replace C by C_{ij} and w by w_{ij} and return to Step 1, taking the sequence of vertices $v_1v_2v_3\dots v_nv_1$ as given by our new C.
5. Set $j = j+1$. If $j < n$, do Step 4. Otherwise, set $i = i+1$. If $i < n-2$, do Step 3. Otherwise, stop.

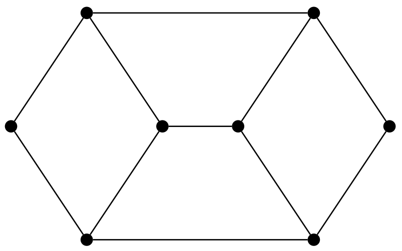
3.9 Exercise Problems

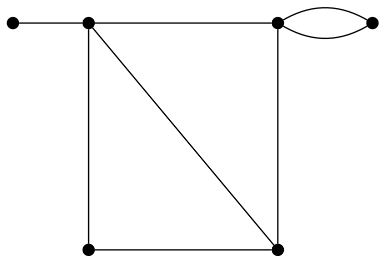
1. Draw a graph that has a Hamilton path, but not a Hamilton Cycle.

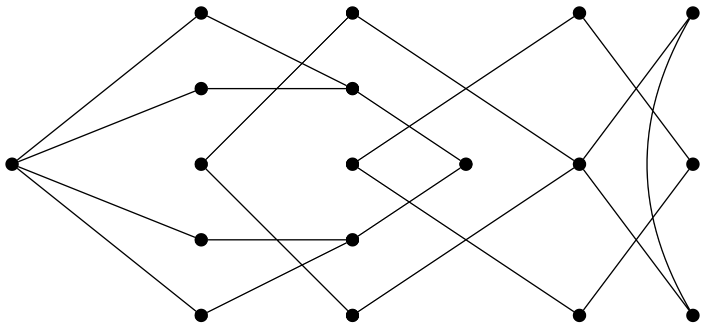












2. Show that if G is Eulerian,then every block (see Chapter 7 for the notion of blocks) of G is Eulerian.
3. Show that a Hamilton path of a graph G, if exists, is the longest path in G.
4. Show that if G is a self-complementary graph, then G has a Hamilton path.
5. Show that every complete graph $K_n; n \geq 3$ is Hamiltonian.
6. Verify whether Petersen graph is Eulerian. Justify your answer.
7. Verify whether Petersen graph is Hamiltonian. Justify your answer.
8. Verify whether the following graphs are Eulerian and Hamiltonian. Justify your answer.
9. Show that every even graph (a graph without odd degree vertices) can be decomposed into cycles.
10. Show that if either
    (a) G is not 2-connected, or
    (b) G is bipartite with bipartition (X,Y) where $|X| \leq |Y|$,
    then G is non-Hamiltonian.
11. Characterise all simple Euler graphs having an Euler tour which is also a Hamiltonian cycle.
12. Let G be a Hamiltonian graph. Show that G does not have a cut vertex.
13. There are n guests at a dinner party, where $n \geq 4$. Any two of these guests know, between them, all the other $n-2$. Prove that the guests can be seated round a circular table so that each one is sitting between two people they know.
14. Let G be a simple k-regular graph, with $2k-1$ vertices. Prove that G is Hamiltonian.

Chapter 4

Directed Graphs

****

4.1 Directed Graphs

Definition 4.1.1 (Directed Graphs)**.** A *directed graph* or *digraph* G consists of a set V of vertices and a set E of edges such that $e \in E$ is associated with an ordered pair of vertices. In other words, if each edge of the graph G has a direction, then the graph is called a directed graph.

The directed edges of a directed graph are also called *arcs*. The initial vertex of an arc a is called the *tail* of a and the terminal vertex v is called the *head* of the arc a. An arc $e = (u,v)$ in a digraph D is a *loop* if $u = v$. Two arcs e, f are *parallel edges* if they have the same tails and the same heads. If D has no loops or parallel edges, then we say that D is *simple*.

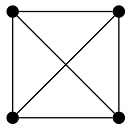
Definition 4.1.2 (Degrees in Digraphs)**.** The *indegree* of vertex v in a directed graph D is the number of edges which are coming into the vertex v (that is, the number of incoming edges) and is denoted by $d^-(v)$. The *out-degree* vertex v in a directed graph D is the number of edges which are going out from the vertex v (that is, the number of outgoing edges) and is denoted by $d^+(v)$.

Definition 4.1.3 (Orientation of Graphs)**.** If we assign directions to the edges of a given graph, then the new directed graph D is called an *orientation* of G.

Definition 4.1.4 (Underlying Graphs of Directed Graphs)**.** If remove the directions of the edges of a directed graph D, then the reduced graph G is called the *underlying graph* of D.

Note that the orientation of a graph G is not unique. Every edge of G can take any one of the two possible directions. Therefore, a graph $G = (V,E)$ can have at most $2^{|E|}$ different orientations. But, a directed graph can have a unique underlying graph.

Figure 4.1b illustrates an undirected graph G and an orientation D of G.


(a) A graph G

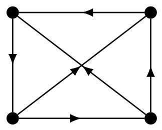

(b) Orientation of G

Figure 4.1: An undirected graph and one of its orientations.

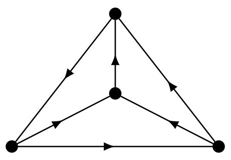
Definition 4.1.5 (Sources and Sinks)**.** A vertex with zero in-degree is called a *source* and a vertex with zero out-degree is called a *sink*.

Theorem 4.1.6. *In a directed graph D, the sum of the in-degrees and the sum of out-degrees of the vertices is equal to twice the number of edges. That is,* $\sum\limits_{v\in V(D)} d^+(v) = \sum\limits_{v\in V(D)} d^-(v) = \epsilon$.

Proof. Let $S^+ = \sum\limits_{v\in V(D)} d^+(v)$ and $S^- = \sum\limits_{v\in V(D)} d^-(v)$. Notice that every arc (edge) of D contributes exactly 1 to S^+ and 1 to S^-. That is, we count each edge exactly once in S^+ and once in S^-. Hence, $S^+ = S^- = |E| = \epsilon$. □

Definition 4.1.7 (Tournament)**.** If the edges of a complete graph are each given an orientation, the resulting directed graph is called a *tournament*.

Definition 4.1.8 (Complete Digraph)**.** A *complete digraph* is a directed graph in which every pair of distinct vertices is connected by a pair of unique edges (one in each direction).


(a) A tournament on 4 vertices

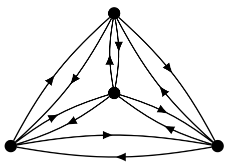

(b) A complete digraph on 4 vertices

Figure 4.2: A tournament and a complete digraph

Definition 4.1.9 (Paths and Cycles in Directed Graphs)**.** A *directed walk* in a digraph D is a sequence $v_0, e_1, v_2, \ldots, e_n, v_n$ so that $v_i \in V(D)$ for every $0 \le i \le n$, and e_i is an edge from v_{i-1} to v_i for every $1 \le i \le n$. We say that this is a walk from

v_0 to v_n. If $v_0 = v_n$, then the walk is called a *closed directed walk* and if $v_0, v_1, \ldots, v_n$ are distinct we call it a *directed path*. In a directed path $v_0, e_1, v_2, \ldots, e_n, v_n$, if $v_0 = v_n$, then the directed path is called a *directed cycle*.

Definition 4.1.10 (Weakly and Strongly Connected Digraphs)**.** A digraph D is said to be *weakly connected* if its underlying graph G is connected. A digraph D is said to be *strongly connected* if there is a directed path between any two vertices $u, v \in V(D)$.

Some observations on directed graphs are given below.

1. Let D be a digraph in which every vertex has outdegree at least 1. Then D contains a directed cycle.
2. A digraph D is *acyclic* if it has no directed cycles. The digraph D is acyclic if and only if there is an ordering $v_1, v_2, \ldots, v_n$ of $V(D)$ so that every edge (v_i, v_j) satisfies $i < j$.

Definition 4.1.11 (Eulerian Digraphs)**.** A closed directed walk in a digraph D is called *Eulerian* if it contains every edge exactly once. A digraph D is said to be an *Eulerian digraph* if it contains an Eulerian closed directed walk.

The following theorem describes a necessary and sufficient condition for a digraph to be Eulerian.

Theorem 4.1.12. *A digraph D, whose underlying graph is connected, is Eulerian if and only if $d^+(v) = d^-(v)$ for every $v \in V(D)$.*

Proof. (*Necessary Part:*) First assume that D is Eulerian. Then, D consists of an Eulerian closed directed walk. That is, whenever there is an incoming edge to v, then there is an outgoing edge from v. Therefore, $d^+(v) = d^-(v)$ for every $v \in V(D)$.
(*Converse Part:*) Choose a closed walk $v_0, e_1, v_1, e_2, v_2 \ldots, e_n, v_n$ which uses each edge at most once and is maximum in length, subject to the given constraint. Suppose that this walk is not Eulerian. Then, as in the undirected case, it follows from the fact that the underlying graph is connected that there exists an edge $e \in E(D)$ which does not appear in the walk so that e is incident with some vertex in the walk, say v_i. Let $H = D - \{e_1, e_2, \ldots, e_n\}$. Then, every vertex of H has indegree equal to its outdegree, so by the previous proposition, there is a list of directed cycles in H containing every edge exactly once. In particular, there is a directed cycle $C \subseteq H$ with $e \in C$. But then, the walk obtained by following $v_0, e_1, \ldots, v_i$, then following the directed cycle C from v_i back to itself, and then following $e_{i+1}, v_i, \ldots, v_n$ is a longer closed walk which contradicts our choice. This completes the proof. $\square$

4.2 Relations and Directed graphs

Digraph are the most common ways to represent different kinds of binary relations defined on a set X. Each element x in X is represented as a vertex in the corresponding digraph and if an element $x \in X$ is related to another element $y \in X$, then there is a directed edge from the vertex x to the vertex v in the corresponding digraph.

For example, Figure 4.3 represents the relation "less than" defined on the set $X = \{1,2,3,4\}$.

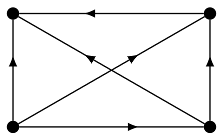


Figure 4.3: A relation digraph

Remark 4.2.1. Note that every binary relation on a finite set can be represented by a digraph without parallel edges. The converse of this statement is also true. That is, every (finite) digraph without parallel edges defines a binary relation on the set of its vertices.

4.2.1 Types of Digraphs

Definition 4.2.2 (Reflexive Digraphs)**.** *Reflexive digraphs* are the directed graphs with self-loops at all vertices. A directed graph in which no vertex has self-loop is called *irreflexive digraphs.*

Figure 4.4 depicts a reflexive digraph.

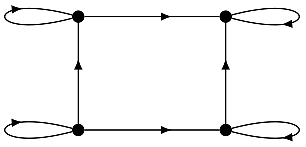


Figure 4.4: A reflexive digraph

Definition 4.2.3 (Symmetric Digraphs)**.** *Symmetric directed graphs* are the directed graphs where all edges are bidirected (that is, for every arrow that belongs to the digraph, the corresponding inversed arrow also belongs to it). In other words, Symmetric digraphs are the digraphs in which every edge (a,b), there exists an edge (b,a) also.

Note that, symmetric digraphs are the graphical representations of symmetric relations. See Figure 4.5 for an illustration.

Definition 4.2.4 (Asymmetric digraphs)**.** *Asymmetric digraphs* are the digraphs that at most one directed edge between a pair of vertices, but are allowed to have self-loops.

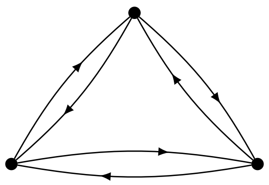


Figure 4.5: A symmetric digraph

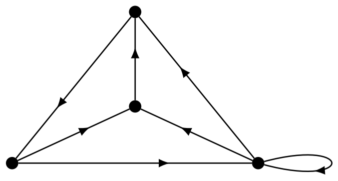


Figure 4.6: An asymmetric digraph

See Figure 4.6 for an illustration.

Definition 4.2.5 (Transitive Digraphs)**.** A digraph D is said to be a *transitive digraph*, if any three vertices (x, y, z) such that edges (x, y) and (y, z) in G imply (x, z) in D. Unlabeled transitive digraphs are called *digraph topologies*.

Figure 4.7 illustrates a transitive digraph.

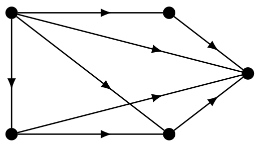


Figure 4.7: A transitive digraph

Definition 4.2.6 (Equivalence Digraphs)**.** An *equivalence digraph* is a directed graph which represents an equivalence relation.

Note that an equivalence digraph is reflexive, symmetric and transitive. An equivalence digraph is depicted in Figure 4.8.

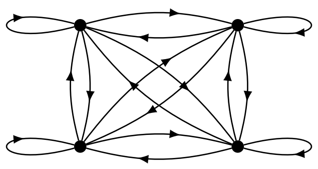


Figure 4.8: A equivalence digraph

Chapter 5

Trees

****

5.1 Trees

Definition 5.1.1 (Tree)**.** A graph G is called a *tree* if it is connected and has no cycles. That is, a tree is a connected acyclic (circuitless) graph.

Definition 5.1.2 (Tree)**.** An acyclic graph may possibly be a disconnected graph whose components are trees. Such graphs are called **forests**.

Note that a forest can be considered as a disjoint union of trees. A pendant vertex of a tree (or a forest) is called its *leaf*.
Illustrations to trees are given in Figure 5.1.

5.2 Properties of Trees

Theorem 5.2.1. *A graph is a tree if and only if there is exactly one path between every pair of its vertices.*

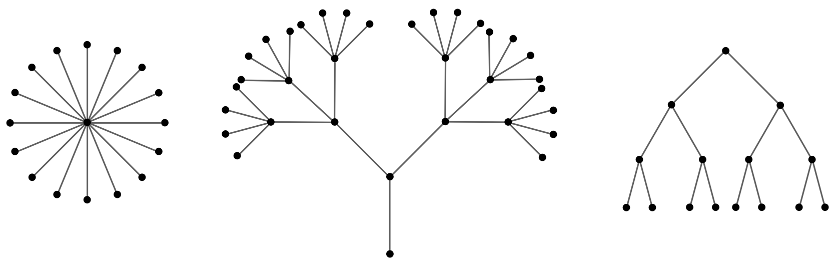


Figure 5.1: Examples of trees

Proof. Let G be a graph and let there be exactly one path between every pair of vertices in G. So G is connected. If G contains a cycle, say between vertices u and v, then there are two distinct paths between u and v, which is a contradiction to the hypothesis. Hence, G is connected and is without cycles, therefore it is a tree.

Conversely, let G be a tree. Since G is connected, there is at least one path between every pair of vertices in G. Let there be two distinct paths, say P and P' between two vertices u and v of G. Then, the union of $P \cup P'$ contains a cycle which contradicts the fact that G is a tree. Hence, there is exactly one path between every pair of vertices of a tree. □

Then, by Definition 2.1.13, we have the following result:

Theorem 5.2.2. *All trees are geodetic graphs.*

Theorem 5.2.3. *A tree with n vertices has $n-1$ edges.*

Proof. We prove the result by using mathematical induction on n, the number of vertices. The result is obviously true for $n = 1,2,3$. See illustrations in Figure 5.2.

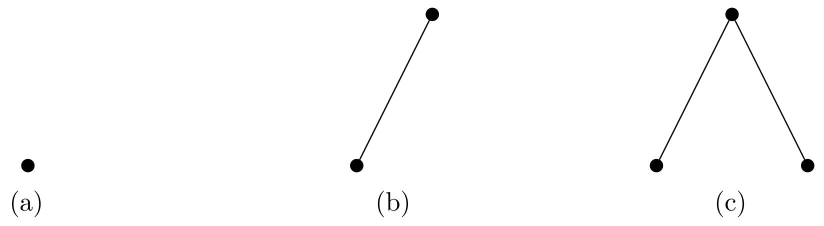


Figure 5.2: Trees with $n = 1,2,3$.

Let the result be true for all trees with fewer than n vertices. Let T be a tree with n vertices and let e be an edge with end vertices u and v. So, the only path between u and v is e. Therefore, deletion of e from T disconnects T.

Now, $T-e$ consists of exactly two components T_1 and T_2 say, and as there were no cycles to begin with, each component is a tree. Let n_1 and n_2 be the number of vertices in T_1 and T_2 respectively. Then, note that $n_1+n_2 = n$. Also, $n_1 < n$ and $n_2 < n$. Thus, by induction hypothesis, the number of edges in T_1 and T_2 are respectively n_1-1 and n_2-1. Hence, the number of edges in T is $n_1-1+n_2-1+1 = n_1+n_2-1 = n-1$. □

Theorem 5.2.4. *Any connected graph with n vertices and $n-1$ edges is a tree.*

Proof. Let G be a connected graph with n vertices and $n-1$ edges. We show that G contains no cycles. Assume to the contrary that G contains cycles. Remove an edge from a cycle so that the resulting graph is again connected. Continue this process of removing one edge from one cycle at a time till the resulting graph H is a tree. As H has n vertices, so the number of edges in H is $n-1$. Now, the number of edges in G is greater than the number of edges in H. That is, $n-1 > n-1$, which is not possible. Hence, G has no cycles and therefore is a tree. □

Theorem 5.2.5. *Every edge of a tree is a cut-edge of G.*

Proof. Since a tree T is an acyclic graph, no edge of T is contained in a cycle. Therefore, by Theorem 2.3.5, every edge of T is a cut-edge. □

Definition 5.2.6 (Minimally Connected Graph)**.** A graph is said to be *minimally connected* if removal of any one edge from it disconnects the graph.

Clearly, a minimally connected graph has no cycles. Then, the following theorem is another characterization of trees.

Theorem 5.2.7. *A graph is a tree if and only if it is minimally connected.*

Proof. Let the graph G be minimally connected. Then, G has no cycles and therefore is a tree. Conversely, let G be a tree. Then, G contains no cycles and deletion of any edge from G disconnects the graph. Hence, G is minimally connected. □

Theorem 5.2.8. *A graph G with n vertices, $n-1$ edges and no cycles is connected.*

Proof. Let G be a graph without cycles with n vertices and $n-1$ edges. We have to prove that G is connected. Assume that G is disconnected. So G consists of two or more components and each component is also without cycles. We assume without loss of generality that G has two components, say G_1 and G_2. Add an edge e between a vertex u in G_1 and a vertex v in G_2. Since there is no path between u and v in G, adding e did not create a cycle. Thus $G \cup \{e\}$ is a connected graph (tree) of n vertices, having n edges and no cycles. This contradicts the fact that a tree with n vertices has $n-1$ edges. Hence, G is connected. □

Theorem 5.2.9. *Any tree with at least two vertices has at least two pendant vertices.*

Proof. Let the number of vertices in a given tree T be n, where $(n > 1)$. So the number of edges in T is $n-1$. Therefore, the degree sum of the tree is $2(n-1)$ (by the first theorem of graph theory). This degree sum is to be divided among the n vertices. Since a tree is connected it cannot have a vertex of zero degree. Each vertex contributes at least 1 to the above sum. Thus, there must be at least two vertices of degree exactly 1. That is, every tree must have at least two pendant vertices. □

Theorem 5.2.10. *Let G be a graph on n vertices. Then, the following statements are equivalent:*

(i) G is a tree.
(ii) G is connected and has $n-1$ edges.
(iii) G is acyclic (circuitless) and has $n-1$ edges.
(iv) There exists exactly one path between every pair of vertices in G.
(v) G is a minimally connected graph.

Proof. The equivalence of these conditions can be established using the results $(i) \implies (ii), (ii) \implies (iii), (iii) \implies (iv), (iv) \implies (v)$ and $(v) \implies (i)$.

Part-$(i) \implies (ii)$: This part states that if G is a tree on n vertices, then G is connected and has $n-1$ edges. Since G is a tree, clearly, by definition of a tree it is connected. The remaining part follows from the result that every tree on n vertices has $n-1$ vertices.

Part-$(ii) \implies (ii)$: This part states that if G is connected and has $n-1$ edges, then G is acyclic and has $n-1$ edges. Clearly, This result follows from the result that a connected graph on n vertices and $n-1$ edges is acyclic.

Part-$(iii) \implies (iv)$: This part states that if G is an acyclic graph on n vertices and has $n-1$ edges, then there exists exactly one path between every pair of vertices in G. By a previous theorem, we have an acyclic graph G on n vertices and $n-1$ edges is connected. Therefore, G is a tree. Hence, by our first theorem, there exists exactly one path between every pair of vertices in G.

Part-$(iv) \implies (v)$: This part states that if there exists exactly one path between every pair of vertices in G, then G is minimally connected. Assume that every pair of vertices in G is connected by a unique path.

Let u and v be any two vertices in G and P be the unique (u,v)-path in G. Let e be any edge in this path P. If we remove the edge from P, then there will be no (u,v)-path in $G-e$. That is, $G-e$ is disconnected. Therefore, G is minimally connected.

Part-$(v) \implies (i)$: This part states that if G is minimally connected, then G is a tree. Clearly, G is connected as it is minimally connected. Since G is minimally connected, removal of any edge makes G disconnected. That is, every edge of G is a cut edge of G. That is, no edge of G is contained in a cycle in G. Therefore, G is acyclic and hence is a tree. □

Theorem 5.2.11. *Let G be a graph with n vertices and k components. Then, G has at most $\frac{1}{2}(n-k)(n-k+1)$ edges.*

Proof. Let G be a graph with n vertices and k components. Let the number of vertices in each of the k components of G be $n_1, n_2, \dots, n_k$ respectively. Then we have,

$$n_1 + n_2 + \ldots + n_k = n; \; n_i \geq 1 \tag{5.1}$$

First, note that any connected graph on n vertices must have at least $n-1$ edges. The proof of the theorem is based on the inequality $\sum\limits_{i=1}^{k} n_i^2 \leq n^2 - (k-1)(2n-k)$, which can be proved as follows.

$$\begin{aligned}
\sum_{i=1}^{k}(n_i-1) &= n-k \\
\left(\sum_{i=1}^{k}(n_i-1)\right)^2 &= (n-k)^2 \\
\sum_{i=1}^{k}(n_i^2-2n_i)+k+\textit{non-negative cross terms} &= n^2+k^2-2nk \\
\sum_{i=1}^{k}n_i^2-2\sum_{i=1}^{k}n_i+k &\leq n^2+k^2-2nk \\
\sum_{i=1}^{k}n_i^2-2n+k &\leq n^2+k^2-2nk \\
\sum_{i=1}^{k}n_i^2 &\leq n^2+k^2-2nk+2n-k \\
\therefore \sum_{i=1}^{k}n_i^2 &\leq n^2-(k-1)(2n-k).
\end{aligned}$$

Hence, we have

$$\sum_{i=1}^{k}n_i^2 \leq n^2-(k-1)(2n-k) \tag{5.2}$$

Now, note that the number edges in K_n is $\frac{n(n-1)}{2}$. Hence, the maximum number of edges in i-th component of G (which is a simple connected graph) is $\frac{n_i(n_i-1)}{2}$. Therefore, the maximum number of edges in G is

$$\begin{aligned}
\sum_{i=1}^{k}\frac{n_i(n_i-1)}{2} &= \sum_{i=1}^{k}\frac{n_i^2-n_i}{2} \\
&= \frac{1}{2}\sum_{i=1}^{k}(n_i^2-n_i) \\
&= \frac{1}{2}\left[\sum_{i=1}^{k}n_i^2-\sum_{i=1}^{k}n_i\right] \\
&\leq \frac{1}{2}\left[n^2-(k-1)(2n-k)-n\right] \text{ (By Eq. (5.1) and Ineq. (5.2))} \\
&= \frac{1}{2}\left[n^2-2nk+k^2+2n-k-n\right] \\
&= \frac{1}{2}\left[n^2-2nk+k^2+n-k\right]
\end{aligned}$$

$$\begin{aligned} &= \frac{1}{2}\left[(n-k)^2+(n-k)\right] \\ &= \frac{1}{2}(n-k)(n-k+1). \end{aligned}$$

□

Problem 5.2.12. Draw a disconnected graph G with 10 vertices and 4 components. Also, calculate the maximum number of edges possible in G.

Solution: The following is a graph with 10 vertices and 4 components.

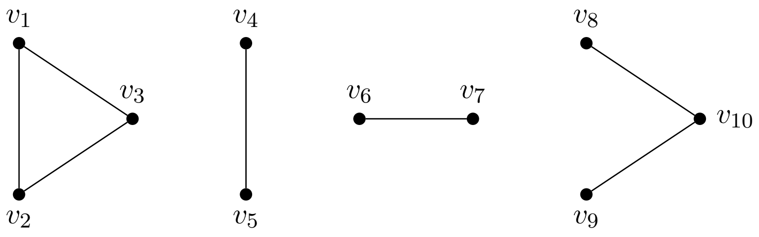


Figure 5.3: G_1

By Theorem 5.2.11, the maximum number of edges of a graph on n vertices with k components is $\frac{1}{2}(n-k)(n-k+1)$. Here, $n=10$ and $k=4$. Therefore, the maximum possible size of G is $\frac{1}{2}(10-4)(10-4+1)=21$. □

Problem 5.2.13. Show that an acyclic graph (graph without cycles) on n vertices and k components has $n-k$ edges.

Solution. The solution follows directly from the first part of the above theorem.

Problem 5.2.14. Show that every graph on n vertices having more than $\frac{(n-1)(n-2)}{2}$ edges is connected.

Solution. Consider the complete graph K_n and v be an arbitrary vertex of K_n. Now remove all $n-1$ edges of K_n incident on v so that it becomes disconnected with K_{n-1} as one component and the isolated vertex v as the second component. Clearly, this disconnected graph has $\frac{(n-1)(n-2)}{2}$ edges (all of which belong to the first component). Since all pairs of vertices in the first component K_{n-1} are any adjacent to each other, any new edge drawn must be joining a vertex in K_{n-1} and the isolated vertex v, making the revised graph connected. □

We have already mentioned that there is no characterisation for a cut-vertex in an arbitrary graph. But we can observe that it is not the case for trees. The following result discusses a necessary and sufficient condition for a vertex of a tree to be its cut vertex.

Theorem 5.2.15. *A vertex v in a tree is a cut-vertex of T if and only if $d(v) \geq 2$.*

Proof. Let v be a cut-vertex of a tree T. Since, no pendant vertex of a graph can be its cut-vertex, clearly we have $d(v) \geq 2$.

Let v be a vertex of a tree T such that $d(v) \geq 2$. Then v is called an *internal vertex* (or *intermediate vertex*) of T. Since $d(v) \geq 2$, there are two at least two neighbours for v in T. Let u and w be two neighbours of v. Then, $u-v-w$ is a $(u-w)$-path in G. By Theorem-1, we have the path $u-v-w$ is the unique $(u-w)$-path in G. Therefore, $T-v$ is disconnected and u and w are in different components of T. Therefore, v is a cut-vertex of T. This completes the proof. □

Hence, the following result is straight forward.

Theorem 5.2.16. *Every internal vertex of a tree is a cut-vertex.*

Proof. An internal vertex of a graph G is a vertex v with degree greater than or equal to 2. Therefore, the result follows from Theorem 5.2.15. □

5.3 Distances in Trees

Definition 5.3.1 (Metric)**.** A *metric* on a set A is a function $d : A \times A \to [0,\infty)$, where $[0,\infty)$ is the set of non-negative real numbers and for all $x,y,z \in A$, the following conditions are satisfied:

1. $d(x,y) \geq 0$ (non-negativity or separation axiom);
2. $d(x,y) = 0 \Leftrightarrow x = y$ (identity of indiscernibles);
3. $d(x,y) = d(y,x)$ (symmetry);
4. $d(x,z) \leq d(x,y) + d(y,z)$ (sub-additivity or triangle inequality).

Conditions 1 and 2, are together called a *positive-definite function.*

A metric is sometimes called the *distance function.*

In view of the definition of a metric, we have

Theorem 5.3.2. *The distance between vertices of a connected graph is a metric.*

Proof. Let u,v and w be any three vertices in a connected graph G. Then, we have

(i) If $u = v$, then $d(u,v) = 0$ and if $u \neq v$, then $d(u,v) > 0$. Therefore, $d(u,v) \geq 0$.
(ii) If P is a shortest path from u to v, then P itself is a shortest path from v to u also. Therefore, $d(u,v) = d(v,u)$.
(iii) Let P_{uv}, P_{uw} and P_{vw} respectively be the shortest uv-path, shortest uw-path and shortest vw-path in the given graph G. If all uw-paths pass through the vertex v, then we have $P_{uw} = P_{uv} \cup P_{vw}$ and hence $d(u,w) = d(u,v) + d(v,w)$. Otherwise, P_{uw} will have smaller length than the path $P_{uv} \cup P_{vw}$. That is, $d(u,w) < d(u,v) + d(v,w)$. Combining these two cases, we have $d(u,w) \leq d(u,v) + d(v,w)$.

Hence, the distance between two vertices in a graph satisfies the three conditions of a metric and hence is a metric. □

Definition 5.3.3 (Center of a graph)**.** A vertex in a graph G with minimum eccentricity is called the *center* of G.

Theorem 5.3.4. *Every tree has either one or two centers.*

Proof. The maximum distance, $\max d(v, v_i)$ from a given vertex v to any other vertex occurs only when v_i is a pendant vertex. With this observation, let T be a tree having more than two vertices. Tree T has two or more pendant vertices.

Deleting all the pendant vertices from T, the resulting graph T' is again a tree. The removal of all pendant vertices from T uniformly reduces the eccentricities of the remaining vertices (vertices in T') by one. Therefore, the centers of T are also the centers of T'. From T', we remove all pendant vertices and get another tree T''. Continuing this process, we either get a vertex, which is a center of T, or an edge whose end vertices are the two centers of T. □

Both types of trees are illustrated in the following figure. In the figure, eccentricity of every vertex in both trees are written near to that vertex and centres are represented by white vertices.

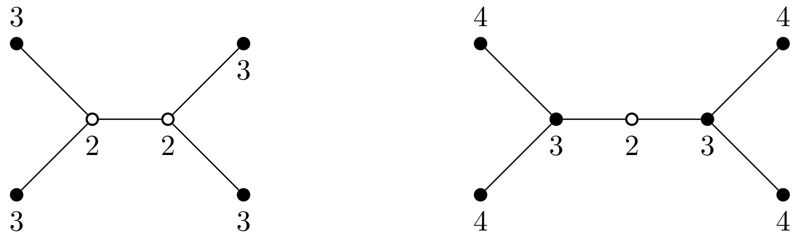


Figure 5.4: Trees with one or two centres.

5.4 Degree Sequences in Trees

Theorem 5.4.1. *The sequence $\langle d_i \rangle; 1 \leq i \leq n$ of positive integers is a degree sequence of a tree if and only if (i) $d_i \geq 1$ for all $i, 1 \leq i \leq n$ and (ii) $\sum d_i = 2n-2$.*

Proof. Since a tree has no isolated vertex, we have $d_i \geq 1$ for all i. Also, $\sum d_i = 2|E| = 2(n-1)$, as a tree with n vertices has $n-1$ edges.

We use induction on n to prove the converse part. For $n = 2$, the sequence is $\{1, 1\}$ and is obviously the degree sequence of K_2 . Suppose the claim is true for all positive sequences of length less than n. Let $\langle d_i \rangle$ be the non-decreasing positive sequence of n terms, satisfying conditions (i) and (ii). Then $d_1 = 1$ and $d_n > 1$.

Now, consider the sequence $D' = \{d_2, d_3, \ldots, d_{n-1}, d_n - 1\}$, which is a sequence of length $n-1$. Obviously in D', we have $d_i \geq 1$ and $\sum d_i = d_2 + d_3 + \ldots + d_{n-1} + d_n - 1 = d_1 + d_2 + d_3 + \ldots + d_{n-1} + d_n - 1 - 1 = 2n - 2 - 2 = 2(n-1) - 2$ (because $d_1 = 1$). So D' satisfies conditions (i) and (ii) and by induction hypothesis, there is a tree T_0 realising D'. In T_0, add a new vertex and join it to the vertex having degree $d_n - 1$ to get a tree T. Therefore, the degree sequence of T is $\{d_1, d_2, \dots, dn\}$. This completes the proof. □

Theorem 5.4.2. *Let T be a tree with k edges. If G is a graph whose minimum degree satisfies $\delta(G) \geq k$, then G contains T as a subgraph. In other words, G contains every tree of order at most $\delta(G)+1$ as a subgraph.*

Proof. We use induction on k. If $k = 0$, then $T = K_1$ and it is clear that K_1 is a subgraph of any graph. Further, if $k = 1$, then $T = K_2$, and K_2 is a subgraph of any graph whose minimum degree is one.

Assume that the result is true for all trees with $k-1$ edges ($k \geq 2$) and consider a tree T with exactly k edges. We know that T contains at least two pendant vertices. Let v be one of them and let w be the vertex that is adjacent to v.

Consider the graph $T - v$. Since $T - v$ has $k-1$ edges, the induction hypothesis applies, so $T - v$ is a subgraph of G. We can think of $T - v$ as actually sitting inside G (meaning w is a vertex of G, too).

Since G contains at least $k+1$ vertices, and $T - v$ contains k vertices, there exist vertices of G that are not a part of the subgraph $T - v$. Further, since the degree of w in G is at least k, there must be a vertex u not in $T - v$ that is adjacent to w. The subgraph $T - v$ together with u forms the tree T as a subgraph of G. □

5.5 Spanning Trees

Definition 5.5.1 (Spanning Tree)**.** A *spanning tree* of a connected graph G is a tree containing all the vertices of G.

A *spanning tree* of a graph is a maximal tree subgraph of that graph. A spanning tree of a graph G is sometimes called the *skeleton* or the *scaffold graph*.

Theorem 5.5.2. *Every connected graph G has a spanning tree.*

Proof. Let G be a connected graph on n vertices. Pick an arbitrary edge of G and name it e_1. If e_1 belongs to a cycle of G, then delete it from G. (Else, leave it unchanged and pick it another one). Let $G_1 = G - e_1$. Now, choose an edge e_2 of G_1. If e_2 belongs to a cycle of G_1, then remove e_2 from G_1. Proceed this step until all cycles in G are removed iteratively. Since G is a finite graph the procedure terminates after a finite number of times. At this stage, we get a subgraph T of G, none of whose edges belong to cycles. Therefore, T is a connected acyclic subgraph of G on n vertices and hence is a spanning tree of G, completing the proof. □

A graph and its spanning tree are given in Figure 5.5.

Problem 5.5.3. Obtain any three non-isomorphic spanning trees of the graph G given below.

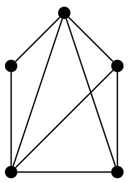
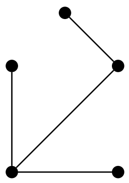
 

Figure 5.5: A graph and one of its spanning trees

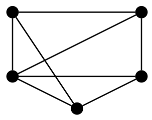


Figure 5.6: Graph G to find spanning trees.

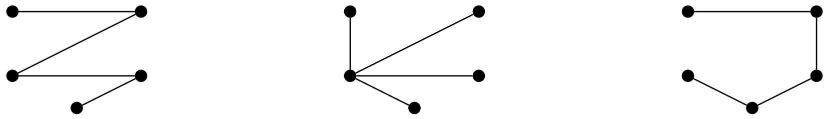


Figure 5.7: Three distinct spanning trees of G

.

Solution: The three distinct spanning trees of G are as given below:

□

Theorem 5.5.4. *Every connected graph on three or more vertices has at least two vertices which are not cut-vertices.*

Proof. Let G be a connected graph of order $n \geq 3$ and let T be a spanning tree of G. Then, by Theorem 5.2.9, there exist at least two pendant vertices in T. Note that no pendant vertex can be a cut-vertex of a graph. Let v be one of the pendant vertices in T. Then, $T - v$ is also a connected (acyclic) graph. Hence, $T - v$ is the spanning tree of the graph $G - v$. Therefore, $G - v$ is connected and hence v is not a cut-vertex of G also. That is, the pendant vertices in a spanning tree of a graph G will not be the cut-vertices of G as well. Therefore, there exist at least two pendant vertices in G. □

Definition 5.5.5 (Branches and Chords of Graphs)**.** Let T be a spanning tree of a given graph G. Then, every edge of T is called a *branch* of T. An edge of G that is not in a spanning tree of G is called a *chord* (or a *tie* or a *link*).

Note that the branches and chords are defined in terms of a given spanning tree.

Theorem 5.5.6. *Show that every graph with n vertices and ϵ edges has $n-1$ branches and $\epsilon - n + 1$ chords.*

Proof. Let G be a graph with n vertices and ϵ edges and let T be a spanning tree of G. Then, by Theorem 5.2.3, T has $n-1$ edges. Therefore, the number of branches is $n-1$. The number chords in G with respect to T is $|E(G)|-|E(T)| = \epsilon-(n-1) = \epsilon-n+1$. □

The set of all chords of a tree T is called a *chord set* or a *co-tree* or a *tie set* and is usually denoted by $\overline{T}$. Therefore, we have $T \cup \overline{T} = G$.

Definition 5.5.7 (Rank and Nullity of Graphs)**.** The number of branches in a spanning tree T of a graph G is called its *rank*, denoted by $rank(G)$. The number of chords of a graph G is called its *nullity*, denoted by $nullity(G)$.

Therefore, we have

$$rank(G) + nullity(G) = |E(G)|, \text{ the size of } G.$$

Theorem 5.5.8. *Let T and T' be two distinct spanning trees of a connected graph G and $e \in E(T)-E(T')$. Then, there exists an edge $e' \in E(T')-E(T)$ such that $T-e+e'$ is a spanning tree of G.*

Proof. By Theorem 5.2.5, we know that every edge of a tree is a cut-edge. Let X_1 and X_2 be the two components of $T-e$. Since T' is a spanning tree and hence T' has an edge e' whose one end vertex in X_1 and the other in X_2. Therefore, the graph $T-e+e'$ is a graph on n vertices and $n-1$ edges and without cycles (since adding an edge between two vertices in two components of G will not create a cycle). Therefore, by Theorem 5.2.8, the graph $T-e+e'$ and hence is a tree. That is, $T-e+e'$ is a spanning tree of G. □

Theorem 5.5.9. *Let T and T' be two distinct spanning trees of a connected graph G and $e \in E(T)-E(T_2)$. Then, there exists an edge $e' \in E(T')-E(T)$ such that $T'+e-e'$ is a spanning tree of G.*

Proof. Since T' is a spanning tree of G and e is not an edge of T', by Theorem 5.7.2, $T'+e$ contains a unique cycle, say C which contains e'. Since T is a tree (hence acyclic), we can find an edge $e' \in E(C)-E(T)$. Therefore, removal of e' breaks the cycle C and hence $T'+e-e'$ is acyclic and has n vertices and $n-1$ edges. Thus, by Theorem 5.2.8, $T'+e-e'$ is connected and hence is a spanning tree of G. □

Theorem 5.5.10 (Cayley's Theorem)**.** *For $n \geq 2$, the number of distinct spanning trees with n vertices is n^{n-2}.*

Proof. Let $V = \{v_1, v_2, \ldots, v_n\}$ be the vertex set of K_n and let $N = \{1, 2, 3, \ldots, n\}$. Note that there exists a bijection $f$from V to N such that $f(v_i) = i$ and V can be replaced by N in this discussion. We note that n^{n-2} is the number of sequences of length $n-2$ that can be formed from N. Thus, in order to prove the theorem, we try to establish a one-to-one correspondence between the set of spanning trees of

K_n and the set of such sequences length $n-2$ from N. With each spanning tree T of K_n, we associate a unique sequence $(t_1,t_2,t_3,\ldots,t_{n-2})$ as follows. Regarding V as an ordered set, let s_1 be the first vertex of degree 1 in T; the vertex adjacent to s_1 is taken as t_1. Let s_2 the first vertex of degree 1 in $T-s_1$, and take the vertex adjacent to s_2 as t_2. This operation is repeated until t_{n-2} has been defined and a tree with just two vertices remains. Note that different spanning trees of K_n determine difference sequences.

Now, note that any vertex v of T occurs $d_T(v)-1$ times in the $(t_1,t_2,t_3,\ldots,t_{n-2})$. Thus, the vertices of degree one in T are precisely those that do not appear in this sequence. Therefore, to reconstruct T from the sequence $(t_1,t_2,\ldots,t_{n-2})$, we proceed as follows: Let s_1 be the first vertex of N not in $(t_1,t_2,\ldots,t_{n-2})$; join s_1 to t_1. Next, let s_2 be the first vertex of $N-\{s_1\}$ not in $(t_2,t_3\ldots,t_{n-2})$, and join s_2 to t_2. Proceed in this way until the $n-2$ edges $s_1t_1,s_2t_2,s_3t_3,\ldots,s_{n-}2t_{n-2}$ have been determined. T is now obtained by adding the edge joining the two remaining vertices of $N-\{s_1,s_2,\ldots,s_{n-2}\}$. We can verify that different sequences we get give rise to different spanning trees of Kn.

We have thus established the desired one-to-one correspondence between the distinct number of spanning trees of K_n and the distinct sequences from N of length $n-2$. Therefore, the number of distinct spanning trees of a graph G is n^{n-2}. □

5.6 On Counting Trees

A *labelled graph* is a graph, each of whose vertices (or edges) is assigned a unique name $(v_1,v_2,v_3,\ldots$ or $A,B,C,\ldots)$ or labels $(1,2,3,\ldots)$.

The distinct vertex labelled trees on 4 vertices are given in Figure 5.8.

The distinct unlabelled trees on 4 vertices are given in Figure 5.9.

Theorem 5.6.1 (Cayley's Theorem on Labelled Trees). *For $n \geq 2$, the number of labelled trees with n vertices is n^{n-2}.*

Proof. Every labelled tree on n vertices can be treated as a spanning tree of K_n. Therefore, the theorem follows immediately from Theorem 5.5.10. □

5.7 Fundamental cycles

Theorem 5.7.1. *A connected graph G is a tree if and only if adding an edge between any two vertices in G creates exactly one cycle (circuit).*

Proof. First assume that G is a tree and let u,v be any two vertices of G. Then, by Theorem 5.2.1, there exists a unique path, say P, between u and v. Add an edge between these two vertices. Then, $P+uv$ is clearly a cycle in the graph $H=G+uv$. If possible, let uv be an edge in two cycles, say C and C' in H. Then, $C-uv$ and $C'-uv$ are two disjoint uv-paths in $H-uv=G$, contradicting the uniqueness of P. Hence, $P+uv$ is the only cycle in $G+uv$.

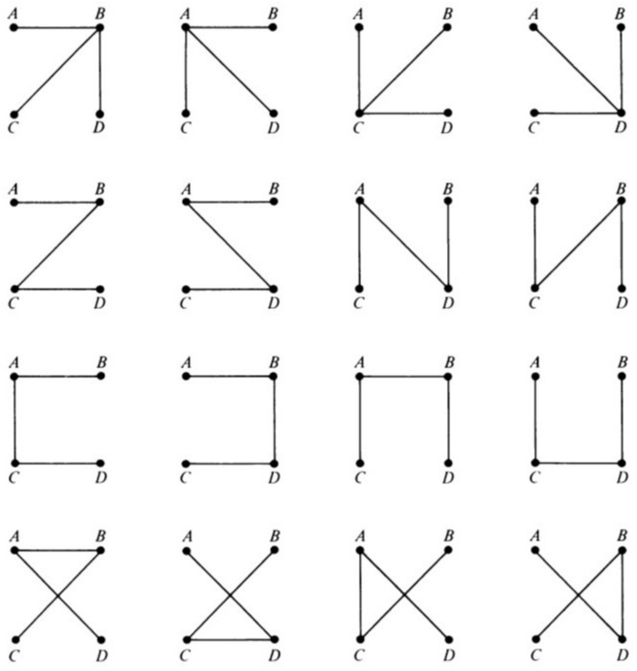


Figure 5.8: Distinct labelled trees on 4 vertices

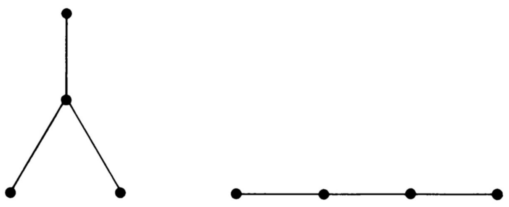


Figure 5.9: Distinct unlabelled trees on 4 vertices

Conversely, assume that $P + uv$ is the only cycle in the graph $H = G + uv$. Then, $G = H - uv$ is a connected graph having no cycles. That is, G is a tree. □

Theorem 5.7.2. *Adding a chord of a connected graph G to the corresponding spanning tree T of G creates a unique cycle and in G.*

Proof. The proof of the theorem is a consequence of Theorem 5.7.1. □

Definition 5.7.3 (Fundamental Cycles)**.** A cycle formed in a graph G by adding a chord of a spanning tree T f G is called a *fundamental circuit* or *fundamental cycle* in G.

Definition 5.7.4 (Cyclomatic Number)**.** The *cyclomatic number* or *circuit rank*, or *cycle rank* of an undirected graph is the minimum number of edges that must be removed from the graph to break all its cycles, making it into a tree or forest.

Clearly, the cyclomatic number of a graph G is equal to the nullity of G.

Theorem 5.7.5. *Any connected graph G with n vertices and ϵ edges has $\epsilon - n + 1$ fundamental cycles.*

Proof. The number of chords corresponding to any spanning tree T of G on n vertices is $\epsilon - n + 1$ (see Theorem 5.5.6). By Theorem 5.7.1, we know that corresponding to each chord in G, there exists a unique fundamental cycle in G. Therefore, the number of fundamental cycles in G is $\epsilon - n + 1$. □

Theorem 5.7.6. *Any connected graph G with n vertices, ϵ edges and k components has $\epsilon - n + k$ fundamental cycles.*

Proof. Any spanning acyclic subgraph F of G with n vertices and k components (may be called a *spanning forest* of G) can have exactly $n - k$ edges (see the first part of Theorem 5.2.11). That is, the number of branches in F is $n - k$. Therefore, the number of chords is $\epsilon - n - k$. Hence, by Theorem 5.7.1, we know that corresponding to each chord in G, there exists a unique fundamental cycle in G. Therefore, the number of fundamental cycles in G is $\epsilon - n + k$. □

Theorem 5.7.7. *If the size of a graph is greater than or equal to its order, then it contains at least one cycle.*

Proof. By Theorem 5.2.10, we know that any connected graph G on n vertices will have at least $n - 1$ edges. If G has $n - 1$ edges, then by Theorem 5.2.10, G is a tree and hence has no cycles. Then, by Theorem 5.7, addition of an edge to G will form exactly one cycle. Hence, any graph of order n and size $m = n + k$, will have at least k cycles, completing the proof. □

Problem 5.7.8. If the size of a connected graph is equal to its order, then show that it contains exactly one cycle.

Solution: The solution is immediate from Theorem 5.7.7. □

If $\{e\}$ is a cut-set of G (That is, when e is a cut edge of G), then it is customary to say that e is a cut-set of G. Hence, we have

Theorem 5.7.9. *Every edge of a tree is a cut-set.*

Proof. The proof is an immediate consequence of the fact that every edge of a tree is a cut-edge of G (see Theorem 5.2.5). □

In view of the above theorem, one can observe that the edge connectivity of T is 1.

Theorem 5.7.10. *Every cut-set in a graph G must contain at least one branch of every spanning tree of G.*

Proof. Let F be a cut-set in G and T be any spanning tree of G. If F does not contain any edge of T, then, by Theorem 5.2.1, there will be a unique path between any pair of vertices in T and hence in $G-F$, contradicting the hypothesis that F is a cut-set in G. Therefore, every cut-set in a graph G must contain at least one branch of every spanning tree of G. □

Theorem 5.7.11. *In any connected graph G, any minimal set of edges consisting of at least one branch of every spanning tree of G is a cut-set.*

Proof. Let F be a minimal set of edges consisting of at least one branch of every spanning tree T of G. Then, $G-F$ will remove at least one edge from every spanning tree of G. Therefore, $G-F$ is disconnected. Since F is minimal, we have $(G-F)+e$ contains a spanning tree of G. Therefore, F is a cut-set of G, completing the proof. □

Problem 5.7.12. Consider the following graph G and any one its spanning trees T. List all fundamental circuits and with respect to T.

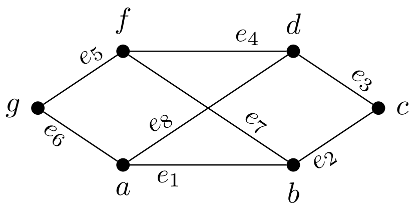


Figure 5.10

Solution: Consider the following spanning tree T of the given graph G obtained by deleting the edges e_6, e_7 and e_8.

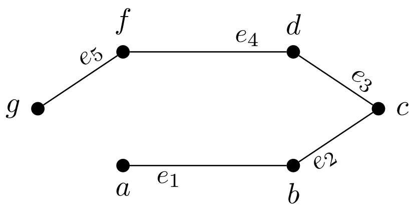


Figure 5.11: A spanning tree T of the graph in Figure 5.10.

The fundamental cycles of G with respect to T are $a e_1 b e_2 c e_3 d e_4 f e_5 g e_6 a$, $b e_2 c e_3 d e_4 f e_7 b$ and $a e_1 b e_2 c e_3 d e_8 a$. □

5.8 Rooted Tree

Definition 5.8.1 (Rooted Tree)**.** A *rooted tree* is a tree T in which one vertex is distinguished from all other vertices. This particular vertex is called the *root* of T.

Figure 5.12 illustrates some rooted trees on five vertices. In all these graphs, the white vertices represent the roots of the rooted trees concerned.

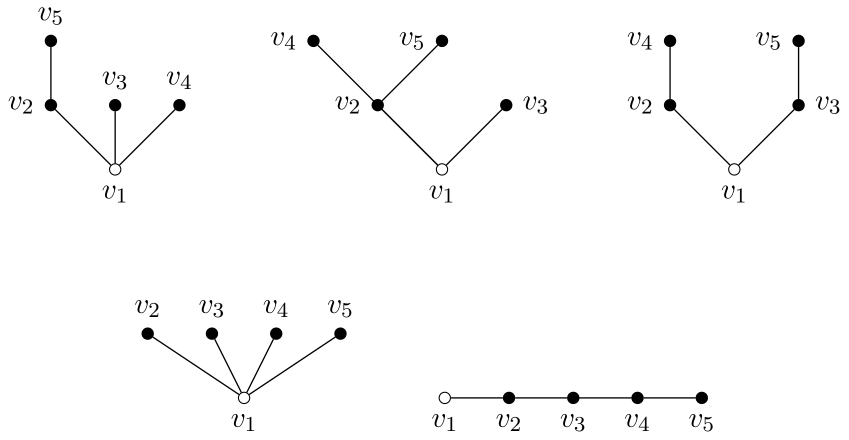


Figure 5.12: Some rooted trees on five vertices

In certain practical or real-life problems, we may have to calculate the lengths or total lengths of vertices of rooted trees from the roots. In some cases, we may also need to assign weights to the vertices of a tree.

Consider the following binary tree whose pendant vertices are assigned some weights.

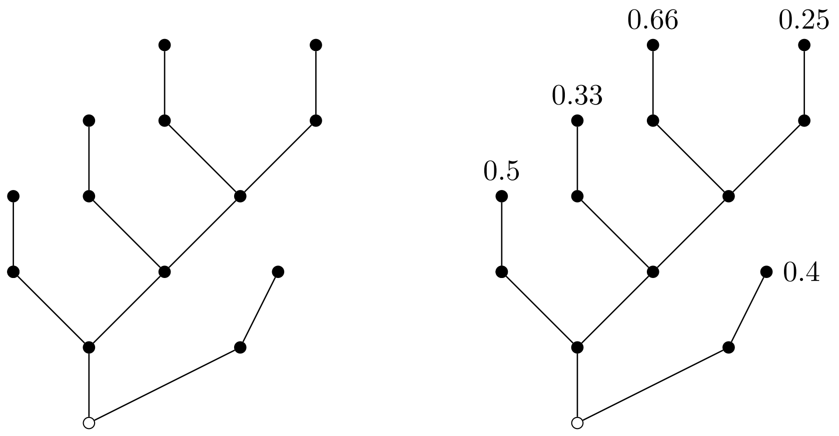


Figure 5.13: A rooted tree with and without weights to pendant vertices.

Definition 5.8.2 (Path Length of a Rooted Tree)**.** The *path length* or (*external path length*) of a rooted tree T is the sum of the levels of all pendant vertices.

The path lengths of rooted trees are widely applied in the analysis of algorithms. The path length of the first rooted tree in Figure 5.13 is $2+3+4+5+5=19$.

Definition 5.8.3 (Weighted Path Length of a Rooted Tree)**.** If every pendant vertex v_i of a tree T is assigned some positive real number w_i, then the *weighted path length* of T is defined as $\sum\limits_i w_i\ell_i$, where ℓ_i is the level of the vertex v_i from the root.

The weighted path length of the graph in Figure 5.13 is $2\times 0.4+3\times 0.5+4\times 0.33+5(0.66+0.25)=8.17$.

5.9 Binary Tree

Definition 5.9.1 (Binary Tree)**.** A *binary tree* is a rooted tree in which there is only one vertex of degree 2 and all other vertices have degree 3 or 1. The vertex having degree 2 serves as the root of a binary tree.

Theorem 5.9.2. *The number of vertices in a binary tree is odd.*

Proof. Let T be a binary tree. Note that the only vertex of T which has even degree is the root. We also have the result that the number of odd degree vertices in any graph is even. Hence, the number of vertices in T is odd. □

Every non-pendant vertex of a binary (or rooted) tree is called its *internal vertex*. A vertex v of a binary tree is said to be *at level* ℓ if its distance from the root is ℓ.
A rooted tree with its levels is illustrated in Figure 5.14.

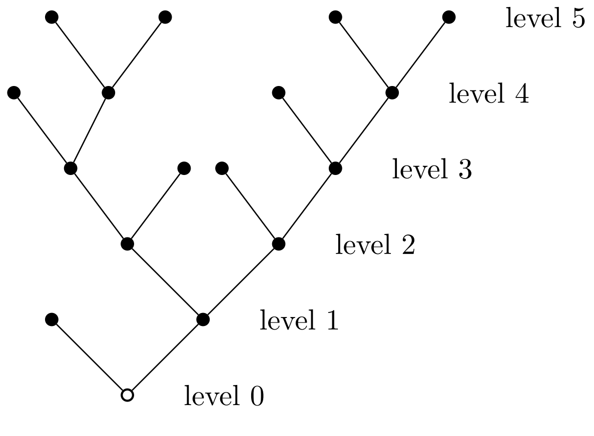


Figure 5.14: A 17-vertex binary tree with level 5.

Theorem 5.9.3. *A binary tree on n vertices has $\frac{n+1}{2}$ pendant vertices.*

Proof. Let p the number of pendant vertices in T. Then, the number of vertices of degree 3 is $n-p-1$. Then, we have

$$\begin{aligned}|E(T)| &= \frac{1}{2}\sum_{v\in V(T)} d(v)(\text{First Theorem on Graph Theory})\\ &= \frac{1}{2}[2+p+3(n-p-1)]\\ &= n-1.\end{aligned}$$

Hence, we have $p = \frac{n+1}{2}$. This complete the proof. □

The binary trees are widely used search procedures. In such procedures, each vertex of a binary tree represents a test and with two possible outcomes. Usually, we have to construct a binary tree on n vertices, for given values of n, with or without a fixed number of levels. This makes the study on the bounds on the number of levels.

Theorem 5.9.4. *Let T be a k-level binary tree on n vertices. Then, $\lceil \log_2(n+1)-1\rceil \le k \le \frac{n-1}{2}$, where $\lceil x\rceil$ represents the smallest integer greater than or equal to x (ceiling function).*

Proof. Note that there is one vertex at level 0 (root), at most two vertices at level 1, at most four vertices at level 2, at most eight vertices at level 3 and proceeding like this, there are at most 2^k vertices at level k.

Figure5.14 is an example for a binary tree with fewer than 2^k vertices at the k-th level for some k, while Figure 5.15 is an example for a binary tree with exactly 2^k vertices at the k-th level for some k.

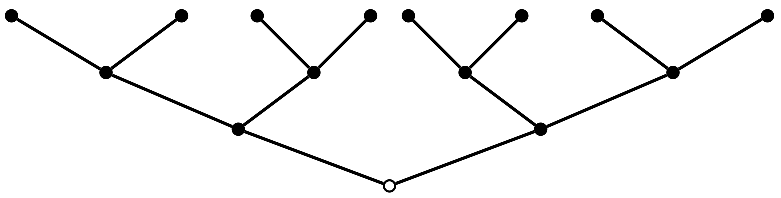


Figure 5.15: A rooted trees 2^k vertices at k-th level, where $k = 0,1,2,3$

Therefore, we have

$$\begin{aligned} n &\le \sum_{i=0}^{k} 2^i\\ &= \frac{2^{k+1}-1}{2-1} \quad (\text{As it is a geometric series with } r=2.)\\ \therefore n &\le 2^{k+1}-1\\ n+1 &\le 2^{k+1}\end{aligned}$$

$$\begin{aligned}\text{That is, } \log_2(n+1) &\leq k+1\\ \text{Therefore, } \log_2(n+1)-1 &\leq k.\end{aligned}$$

Since k is an integer, the above equation becomes $\lceil \log_2(n+1)-1 \rceil \leq k$.

If we construct a binary tree T such that there are exactly two vertices at every level other than the root level, then T attains the maximum possible number of levels (see Figure 5.16).

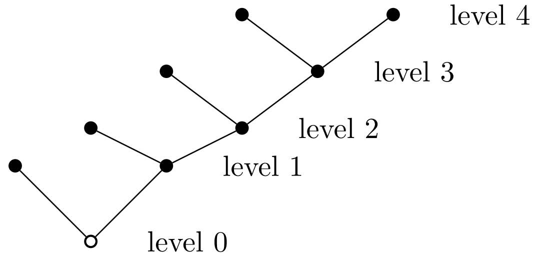


Figure 5.16: A 9-vertex binary tree with level 4.

From this figure, we can interpret that half of the $n-1$ edges of T are drawn towards the left and the remaining half are drawn to the right. Therefore, we can see that the maximum possible value of k is $\frac{n-1}{2}$. □

An ℓ-level binary tree is said to be a *complete binary tree* if it has 2^k vertices at the k-th level for all $1 \leq k \leq \ell$. Figure 5.15 is a complete binary tree of level 3.

5.10 Exercise Problems

1. Show that a path is its own spanning tree.
2. Show that a pendant edge of a graph will be contained in all of its spanning trees.
3. What is the nullity of a complete K_n?
4. Show that every Hamilton path of a graph G (if exists) is a spanning tree of G.
5. Show that every cycle (or circuit) in G has at least one common edge with a chord set.
6. Prove that a graph G is a tree if and only if it is a loopless and has exactly one spanning tree.
7. Every graph with fewer edges than vertices has a component that is a tree.
8. Prove that a maximal acyclic subgraph of G consists of a spanning tree from each component of G.
9. Prove that every graph of order n and size ϵ has at least $\epsilon - n + 1$ cycles.
10. Prove that a simple connected graph having exactly 2 vertices that are not cut-vertices is a path.

11. For a tree T with at least three vertices, show that there is a cut-vertex v of T such that every vertex adjacent to v, except for possibly one, has the degree 1.
12. Let G be connected and let $e \in E$. Then, show that
    (a) An edge e is in every spanning tree of G if and only if e is a cut edge of G;
    (b) e is in no spanning tree of G if and only if e is a loop of G.
13. Show that if each degree in G is even, then G has no cut edge.
14. Show that any tree with at least two vertices is bipartite.
15. With respect to any of its spanning trees, a connected graph of n vertices and m edges has $n-1$ tree branches and $m-n+1$ chords.
16. If T is a tree with $2k \geq 0$ vertices of odd degree, then $E(T)$ is the union of k pair-wise edge-disjoint paths.
17. Let v be any vertex of a connected graph G. Then, show that G has a spanning tree preserving the distances from v.
18. Show that the number of simple, labelled graphs of n vertices is $2^{\binom{n}{2}}$.
19. If T is a spanning tree of a connected graph G and f is a chord of G relative to T , then show that $T+f$ contains a unique cycle of G.
20. Show that any cycle of a connected graph G contains at least one chord of every spanning tree of G.
21. Draw all unlabelled trees with seven and eight vertices.
22. Draw a tree which has radius five and diameter ten.
23. If a tree has an even number of edges, then show that it contains at least one vertex of even degree.
24. If the maximum degree of a vertex in a tree is Δ, then show that it has Δ pendant vertices.
25. Prove that each edge of a connected graph G belongs to at least one spanning tree of G.
26. If it exists, draw a tree whose complement is also a tree.
27. Is it necessary that a graph with n vertices and $n-1$ edges is a tree? Justify your answer.
28. Show that a pendant edge of a graph G will be a part of every spanning tree of G.
29. Every graph with fewer edges than its vertices will contain a tree as its component.

****

Chapter 6

Planar Graphs

****

6.1 Three Utility Problem

Suppose there are three cottages (or houses)- H_1, H_2 and H_3- on a plane and each needs to be connected to three utilities, say gas (G), water (W) and electricity E. Without using a third dimension or sending any of the connections through another house or cottage, is there a way to make all nine connections without any of the lines crossing each other?

The problem can be modelled graphically as follows:

The above graph is often referred to as the utility graph in reference to the problem.

In more formal graph-theoretic terms, the problem asks whether the complete bipartite graph $K_{3,3}$ can be redrawn in such a way that no two edges of $K_{3,3}$ crosses each other.

The three utility problem was the motivation for the introduction of the concepts of planarity of graphs.

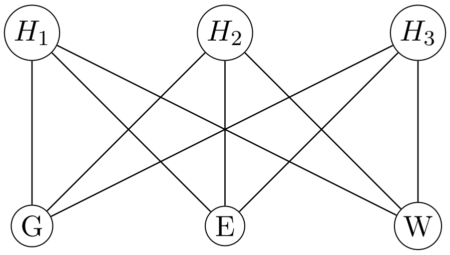


Figure 6.1: Three Utilities Problem

6.2 Planarity of Graphs

Definition 6.2.1. A *face* of a graph G is the region formed by a cycle or parallel edges or loops in G. A face of a graph G is also called a *window*, a *region* or a *mesh*.

Definition 6.2.2 (Jordan Curves)**.** A *Jordan curve* is a non-self-intersecting continuous closed curve in the plane. Cycles in a graph can be considered to be Jordan Curves.

Definition 6.2.3 (Planar graphs)**.** A graph G is called a *planar graph* if it can be re-drawn on a plane without any crossovers (That is, in such a way that two edges intersect only at their end vertices). Such a representation is sometimes called a *topological planar graph.*
Such a drawing of G, if exists, is called a *plane graph* or *planar embedding* or an *embedding* of G.

The portion of the plane lying inside the graph G embedded in a plane is called an *interior region of* G. The portion of the plane lying outside a graph embedded in a plane is infinite in its extent such a region is called an *infinite region*, or *outer region*, or *exterior region* or *unbounded region.*

Every Jordan curve J divides the plane into an *interior region*, denoted by $IntJ$, bounded by the curve and an *exterior region*, denoted by $ExtJ$ containing all of the nearby and far away exterior points.

Definition 6.2.4 (Embedding of a graph)**.** An *embedding* of a graph G on a surface S is a geometric representation of G drawn on the surface such that the curves representing any two edges of G do not intersect except at a point representing a vertex of G.

In order that a graph G is planar, we have to show that there exists a graph isomorphic to G that is embedded in a plane. Equivalently, a geometric graph G is *nonplanar* if none of the possible geometric representations of G can be embedded in a plane.
The graphs depicted in Figure 6.2 are some examples of planar graphs (or plane graphs).

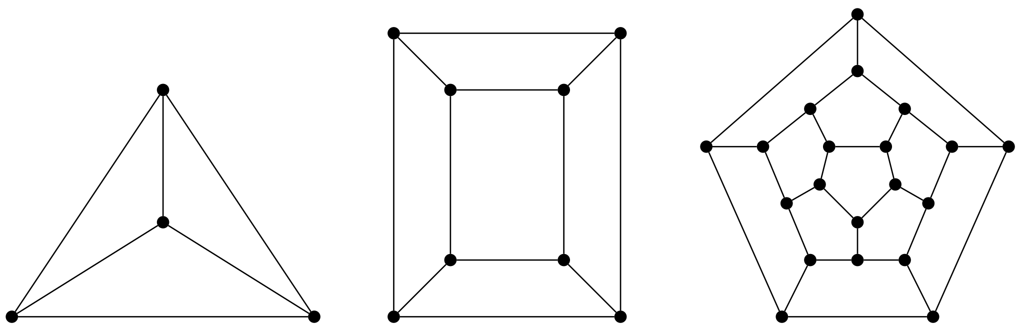


Figure 6.2: Illustration to planar (plane) graphs

A graph which is not planar may be called a *nonplanar graph*.

Definition 6.2.5 (Intersection Number of a Graph)**.** The *intersection number* of a graph G is the minimum number edges of G which intersect other edges, when trying to embed G to a plane.

Note that the intersection number of a planar graph is 0.

Theorem 6.2.6. *A graph can be embedded on the surface of a sphere if and only if it can be embedded in a plane.*

Proof. Consider the stereographic projection of a sphere on the plane. Put the sphere on the plane and call the point of contact as SP (*south-pole*). At point SP, draw a straight line perpendicular to the plane, and let the point where this line intersects the surface of the sphere be called NP (*north-pole*).

Now, corresponding to any point p on the plane, there exists a unique point p_0 on the sphere, and vice versa, where p_0 is the point where the straight line from point p to point NP intersects the surface of the sphere. Thus, there is a one-one correspondence between the points of the sphere and the finite points on the plane, and points at infinity in the plane corresponding to the point NP on the sphere (see Figure 6.3).

Therefore, from this construction, it is clear that any graph that can be embedded in a plane can also be embedded on the surface of the sphere, and vice versa.

The converse can be established as follows: Let P be a point on the sphere, which is placed on the plane. The one-to-one correspondence of the points of the sphere and the points of the plane can be established as the line segment drawn between the north pole N and the point P can be extended till it meets the plane at a unique point $\hat{P}$ on the plane. Therefore, every graph embedded on a sphere can be embedded on a plane also. □

The above theorem can also be stated as given below:

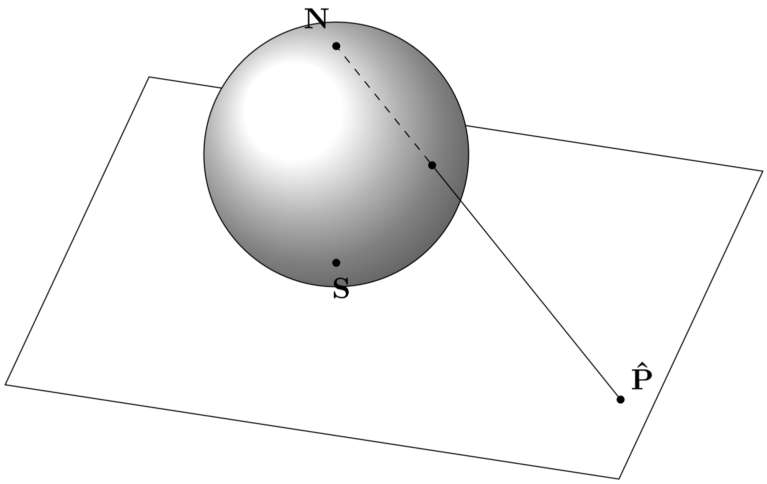


Figure 6.3: Spherical embedding of a point in a plane.

Theorem 6.2.7. *A graph G is planar if and only if it can be embedded on a sphere.*

The embedding of a graph on a sphere is called a *spherical embedding* of a graph.

Note that a planar graph has a unique planar embedding. The following theorem explains how different planar embeddings of a given planar graph can be drawn.

Theorem 6.2.8 (Face-switch Theorem)**.** *A planar embedding G' of a graph G can be transformed into another embedding such that any specified face becomes the exterior face.*

Proof. Any face of G' is defined by the path which forms its boundary. Any such path, T, identified in a particular planar representation P of G, may be made to define the exterior face of a different planar representation P' as follows. We form a spherical embedding P'' of P. P' is then formed by projecting P'' onto the plane in such a way that the point of projection lies in the face defined by the image of T on the sphere.

□

Thinking in terms of the regions on the sphere, it can be noted that there is no real difference between the infinite region and the finite regions on the plane. Therefore, we include the infinite region in our discussions about the regions in a plane representation of the graph.

Also, since there is no essential difference between an embedding of a planar graph on a plane, or on a sphere (a plane can be regarded as the surface of the sphere of infinitely large radius), the term plane representation of a graph is often used to include spherical as well as plane embedding.

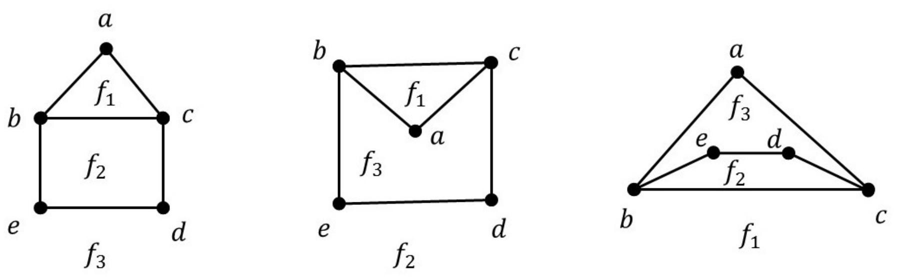


Figure 6.4: The different plane representations of the same graph.

Theorem 6.2.9 (Fary's Theorem)**.** *Every triangulated planar graph has a straight line representation.*

Proof. We prove the result by mathematical induction on n, the number of vertices. The result is obvious for $n = 4$. So, let $n \geq 5$ and assume that the result is true for all planar graphs with fewer than n vertices. Let G be a plane graph with n vertices.

First, we show that G has an edge e belonging to just two triangles. For this, let x be any vertex in the interior of a triangle T and choose x and T such that the number of regions inside T is minimal. Let y be a neighbour of x, and the edge xy lies inside T, and let xy belong to three triangles xyz_1, xyz_2 and xyz_3. Then one of these triangles lies completely inside another. Assume that z_3 lies inside xyz_1. Then, z_3 and xyz_1 contradict the choice of x and T (see Figure 6.5).

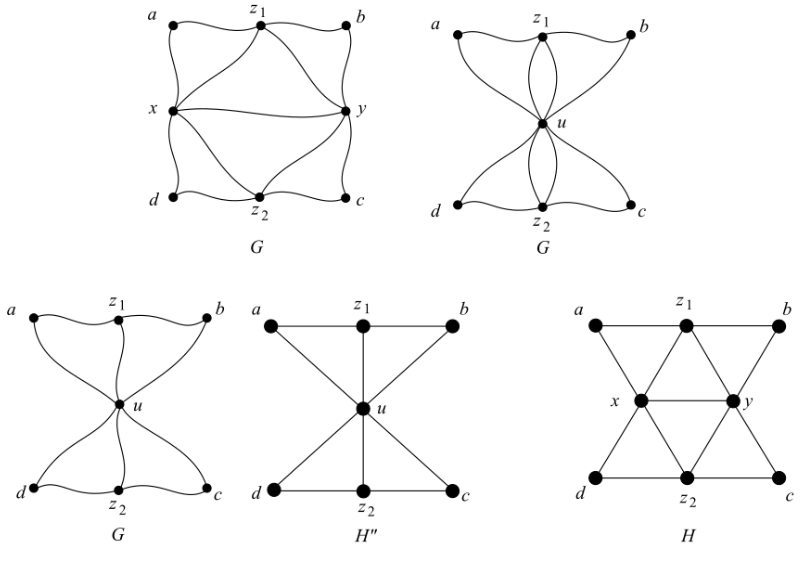


Figure 6.5: Illustration to Fary's Theorem

Hence, there is an edge $e = xy$ lying in just two triangles xyz_1 and xyz_2.

Contracting xy to a vertex u, we get a new graph G' with a pair of double edges between u and z_1, and u and z_2. Remove one each of this pair of double edges to get a graph G'' which is a triangulated graph with $n-1$ vertices. By the induction hypothesis, it has a straight line representation H''. The edges of G'' correspond to uz_1, uz_2 in H''. Divide the angle around u into two parts - in one of which the pre-images of the edges adjacent to x in G lie, and in the other, the pre-images of the edges adjacent to y in G. Hence, u can be pulled apart to x and y, and the edge xy is restored by a straight line to get a straight line representation of G. $\square$

Note that every disconnected graph is planar if and only if each of its components is planar. Similarly, a separable graph is planar if and only if each of its blocks is planar. Hence, one needs to consider non-separable graphs only in questions on planarity.

Theorem 6.2.10 (Jordan Curve Theorem)**.** *The Jordan curve theorem asserts that any path connecting a point of the interior region of a Jordan curve J to a point in the exterior region of J intersects with the curve J somewhere.*

6.3 Kuratowski Graphs and Their Nonplanarity

The complete graph K_5 and the complete bipartite graph $K_{3,3}$ are called *Kuratowski's graphs*, after the Polish mathematician *Kasimir Kuratowski*, who found that these two graphs are nonplanar.

Theorem 6.3.1. *The complete graph with five vertices (K_5) is nonplanar.*

Proof. Let the five vertices in the complete graph be named v_1, v_2, v_3, v_4, v_5. Since every vertex in a complete graph is adjacent to all other vertices by means of an edge, there exists a cycle $C_5 := v_1 - v_2 - v_3 - v_4 - v_5 - v - 1$ (that is, a pentagon) in K_5. This pentagon divides the plane of the paper into two regions, one inside and the other outside (see Figure 6.6a). Since vertex v_1 is to be connected to v_3 and v_4 by means of edges, these edges may be drawn inside or outside the pentagon (without intersecting the five edges drawn previously). Suppose we choose to draw the lines from v_1 to v_3 and v_4 inside the pentagon, (see Figure 6.6b). In case we choose outside, we end with the same argument. Now, we have to draw an edge from v_2 to v_4 and another from v_2 to v_5 . Since neither of these edges can be drawn inside the pentagon without crossing over the edge already drawn, we draw both these edges outside the pentagon, as seen in Figure 6.6c. The edge connecting v_3 and v_5 cannot be drawn outside the pentagon without crossing the edge between v_2 and v_4 . Therefore, v_3 and v_5 have to be connected with an edge. Now, we have to draw an edge between v_3 and v_5 and this cannot be placed inside or outside the pentagon without a crossover, by Jordan Curve Theorem.

Thus, the graph cannot be embedded in a plane. That is, K_5 is non-planar. $\square$

In a similar way, we can prove the non-planarity of the complete bipartite graph $K_{3,3}$ also.

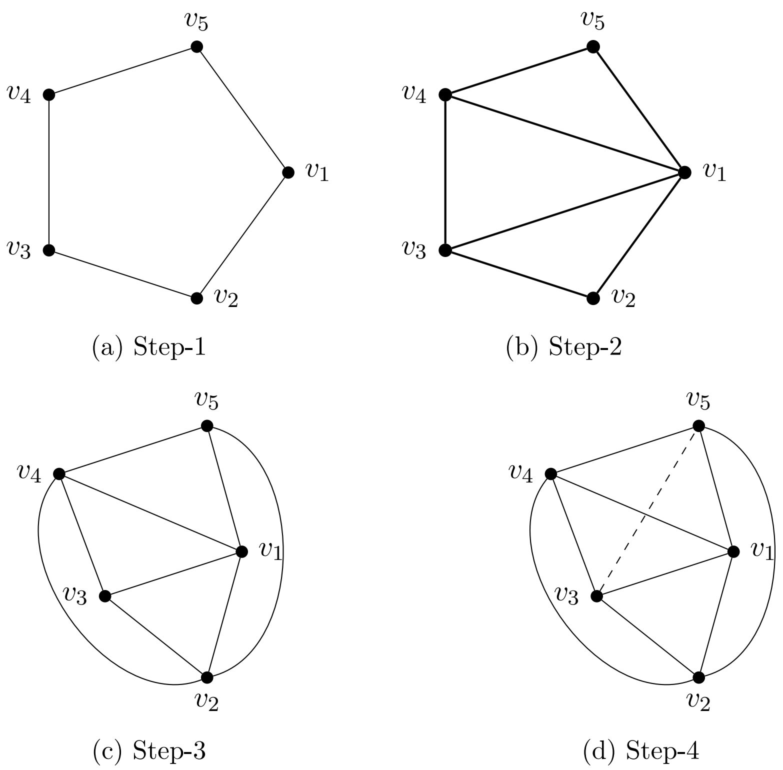


Figure 6.6: Step by step illustration for non-planarity of K_5

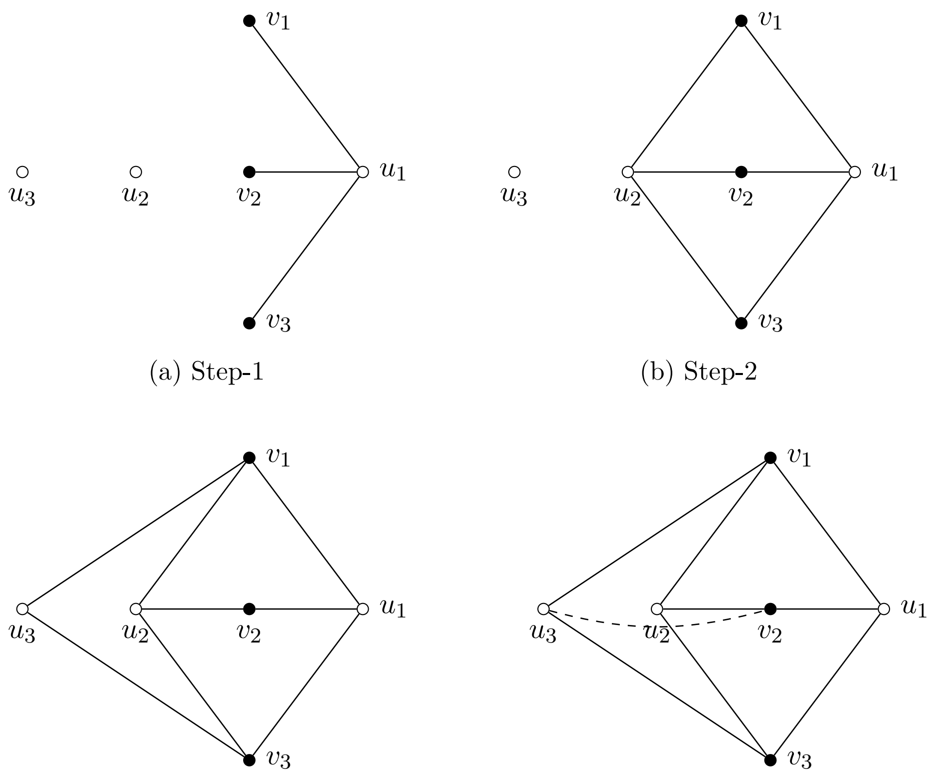
Theorem 6.3.2. *The complete bipartite graph $K_{3,3}$ is nonplanar.*

Proof. The complete bipartite graph $K_{3,3}$ has 6 vertices and 9 edges. Let the vertices be $u_1, u_2, u-3, v_1, v_2, v_3$. We have edges from every u_i to each v_i, $1 \leq i \leq 3$. First, we take the edges from u_1 to each v_1, v_2 and v_3. Then, we take the edges between u_2 to each v_1, v_2 and v_3, as seen in Figure 6.7(a). Thus we get three regions namely I, II and III. Finally, we have to draw the edges between u_3 to each v_1, v_2 and v_3. We can draw the edge between u_3 and v_3 inside the region II without any crossover, (see Figure 6.7(b)). But, the edges between u_3 and v_1, and u_3 and v_2 drawn in any region have a crossover with the previous edges. Thus, the graph cannot be embedded in a plane. Hence, $K_{3,3}$ is nonplanar. □

Remark 6.3.3. Note that exactly one edge each in K_5 and $K_{3,3}$ intersects with other edges while attempting to embed the graphs on plane. The intersection number of the two Kuratwski's graphs K_5 and $K_{3,3}$ is 1.

We observe that the two graphs K_5 and $K_{3,3}$ have the following common properties:

(i) Both K_5 and $K_{3,3}$ are regular.
(ii) K_5 and $K_{3,3}$ are nonplanar.



(a) Step-1 (b) Step-2

(c) Step-3 (d) Step-4

Figure 6.7: Step by step illustration for non-planarity of $K_{3,3}$

(ii) Removal of one edge or a vertex makes each of K_5 and $K_{3,3}$ a planar graph.

(iv) K_5 is a nonplanar graph with the smallest number of vertices, and $K_{3,3}$ is the nonplanar graph with the smallest number of edges.

In view of these facts, we note that K_5 and $K_{3,3}$ are the simplest nonplanar graphs.

6.4 Detection of Planarity and Kuratowski's Theorem

Determining whether a given graph is planar by drawing its plane graph (embedding it to a plane) may not be a feasible method in all cases. So, a new procedure called *elementary topological reduction* or simply, an elementary reduction on a given graph to determine whether it is planar. This process has the following steps:

Elementary Reduction in a graph

Let G be a separable graph with blocks $G_1, G_2, G_3, \ldots, G_k$. Then, we need to check the planarity of each block separately.

S1: Note that addition and/or removal of self-loops do not affect planarity. So, if G has self-loops, remove all of them.

S2: Similarly, parallel edges do not affect planarity. So, if G has parallel edges, remove all of them, keeping one edge between every pair of vertices.

S3: We observe that removal of vertices having degree 2 by merging the two edges incident on it, perform this action as far as possible.

Repeated performance of these steps will reduce the order and size of the graph without affecting its planarity (see Figure 6.8).

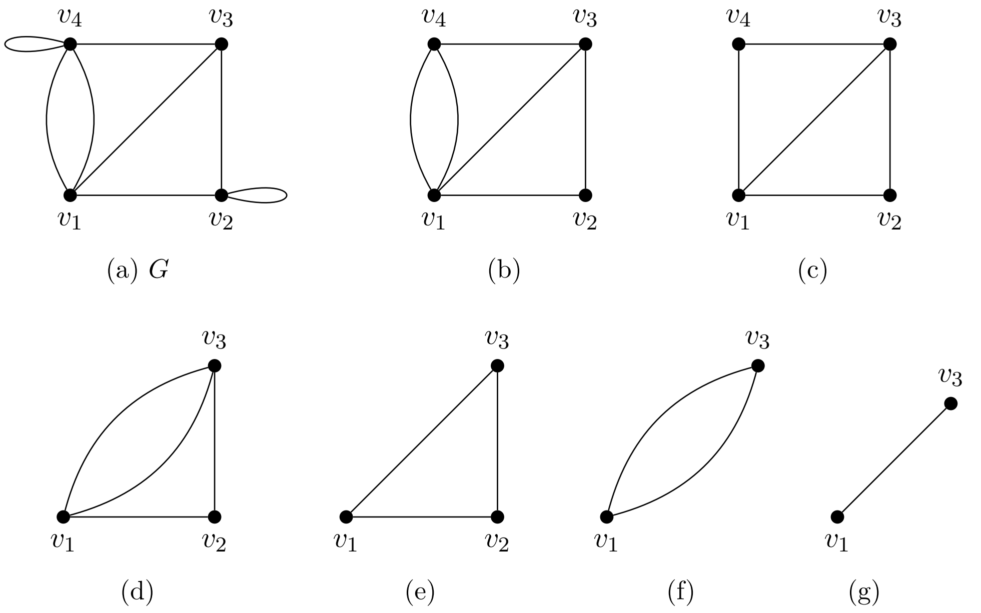


Figure 6.8: Step by step illustration of elementary reduction of a graph.

After repeated application of elementary reduction, the given graph will be reduced to any one of the following cases,

(i) A single edge K_2; or

(ii) A complete graph K_4; or

(iii) A non-separable simple graph with $n \geq 5, \epsilon \geq 7$.

If H is a graph obtained from a graph G by a series of elementary reductions, then G and H are said to be *homeomorphic graphs*. In this case, H is also called a *topological minor* of G.

Lemma 6.4.1. *A graph is planar if and only if its subdivisions are planar.*

Theorem 6.4.2 (Kuratowski's Theorem)**.** *A graph G planar if and only if it has no subdivisions of K_5 and $K_{3,3}$. (*In other words*, a graph G planar if and only if no component of G is homeomorphic to K_5 and $K_{3,3}$).*

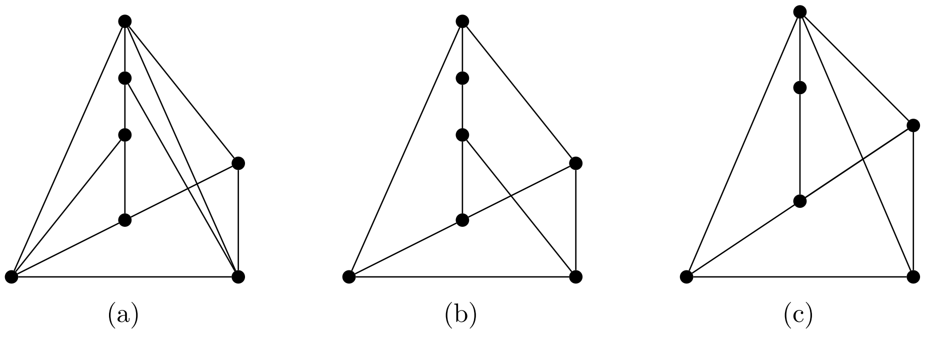


Figure 6.9: Non-planar graph with subgraphs homeomorphic to $K_{3,3}$

6.5 Euler Theorem and Consequences

Theorem 6.5.1 (Euler Theorem on Plane Graphs)**.** *If G is a connected plane graph, then $|V(G)|+|F(G)|-|E(G)|=2$, where V, E, F are respectively the vertex set, edge set and set of faces of G.*

Proof. Let V, E, F be the sets of vertices, edges and faces of a plane graph G. Let $n=|V|, f=|F| and \epsilon=|E|$. We prove the result by mathematical induction on $|E|$. If $\epsilon=0$, then $n=1$ and $f=1$. Also, if $\epsilon=1$, then $n=2$ and $f=1$. In this case also, we have $n+f-\epsilon=2$.

Now, assume that the theorem holds true for all connected graphs with fewer than $\epsilon \geq$ edges, and let G be a connected plane graph with ϵ edges. If G is a tree, then $n=\epsilon+1$ and $f=1$. Hence, we have $n+f-\epsilon=\epsilon+1+1-\epsilon=2$, implying the result.

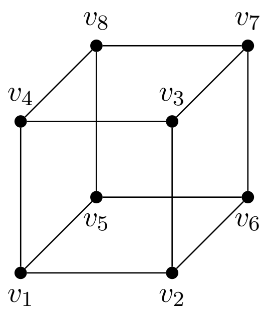
If G is not a tree, then it has an enclosed face. The edges of the face form a cycle. Take any edge e on the cycle and consider graph $H=G-e$. Since $|E(H)|=|E(G)|-1=\epsilon-1<\epsilon$, by induction, $|V(H)|+|F(H)|-|E(H)|=2$. But we know that $|E(H)|=\epsilon-1, |V(H)|=|V(G)|=n$ and $|F(H)|=|F(G)|-1=f-1$.

Therefore,

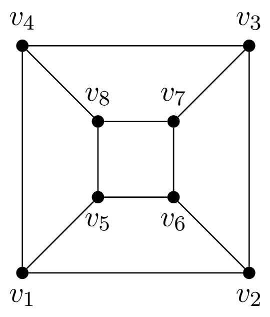
$$\begin{aligned} |V(H)|+|F(H)|-|E(H)| &= 2 \\ \implies n+f-1-(\epsilon-1) &= 2 \\ \implies n+f-1-\epsilon+1 &= 2 \\ \implies n+f-\epsilon &= 2. \end{aligned}$$

Hence, the result follows by mathematical induction. □

In three-dimensional space, a *platonic solid* is a regular, convex polyhedron. Rectangular prisms, cubes, octahedron, dodecahedron etc. are some examples. Note that every polyhedron can be embedded into a plane (see Figure 6.10for example). That is, the graphs corresponding to different polyhedra are planar. Therefore, the above theorem is also known as *Euler's theorem on platonic bodies* or *Euler's theorem on polyhedra.*



(a) A cube



(b) Graph of a cube

Figure 6.10: A cube and its graphical representation

Theorem 6.5.2. *If G is a planar graph without parallel edges on n vertices and ϵ edges, where $\epsilon \geq 3$, then $\epsilon \leq 3n-6$. Moreover, if G is bipartite, then $\epsilon \leq 2n-4$.*

Proof. Let f be the number of faces of G and let m_i be the number of edges in the boundary of the i-th face, where $i = 1,2,\ldots,f$. Since every face contains at least three edges, we have $m_i \geq 3$ for all $i = 1,2,3,\ldots,f$. Then,

$$3f \leq \sum_{i=1}^{f} m_i. \tag{6.1}$$

On the other hand, since every edge can be on the boundary of at most two faces, we have

$$\sum_{i=1}^{f} m_i \leq 2\epsilon. \tag{6.2}$$

From equations (1) and (2), we have $3f \leq 2\epsilon$ or $f \leq \frac{2}{3}\epsilon$. Now, by Euler's theorem, we have

$$\begin{aligned} n+f-\epsilon &= 2 \\ \implies n+\frac{2}{3}\epsilon-\epsilon &\leq 2 \\ \implies n-\frac{\epsilon}{3} &\geq 2 \\ \implies \frac{\epsilon}{3} &\leq n-2 \\ \implies \epsilon &\leq 3n-6. \end{aligned}$$

Part-(II) If G is bipartite, the shortest cycle is of length at least 4. Thus, we have

$$4f \leq \sum_{i=1}^{f} m_i \tag{6.3}$$

As mentioned in the previous case, we also have

$$\sum_{i=1}^{f} m_i \leq 2\epsilon. \tag{6.4}$$

From (3) and (4), we have $4f \leq 2\epsilon$ or $f \leq \frac{1}{2}\epsilon$. By Euler Theorem, we have,

$$\begin{aligned} n+f-\epsilon &= 2 \\ \implies n+\frac{1}{2}\epsilon-\epsilon &\leq 2 \\ \implies n-\frac{\epsilon}{2} &\geq 2 \\ \implies \frac{\epsilon}{2} &\leq n-2 \\ \implies \epsilon &\leq 2n-4. \end{aligned}$$

This completes the proof. □

Theorem 6.5.3. *In a maximal planar graph of order at least* 4*, the minimum degree is at least* 3*.*

Proof. Let G be the corresponding graph with size ϵ. Then, by Theorem 6.5.2, $\epsilon \leq 3n-6$. Since G is maximal planar, G will not have any isolated or pendant vertices. MOreover, $\epsilon = 3n-6$. We will show that G contains no vertex of degree 2. Suppose the contrary, and let $v \in V(G)$ for which $d(v) = 2$. Then $G-v$ is a planar graph of order $n-1$ and size $\epsilon-2$. Since $\epsilon = 3n-6$, we get $\epsilon-2 = 3n-8 = 3(n-1)-5 > 3(n-1)-6$, a contradiction to Theorem 6.5.2. This completes the proof. □

Theorem 6.5.4. *The complete graph K_5 is non-planar.*

Proof. If possible, let K_5 be a planar graph. Then, by above theorem, $\epsilon \leq 3n-6$. In K_5, we have $n = 5$ and $\epsilon = 10$. Hence $3n-6 = 9 < \epsilon = 10$, which contradicts the previous result. Hence, K_5 is non-planar. □

Theorem 6.5.5. *The complete bipartite graph $K_{3,3}$ is non-planar.*

Proof. If possible, let $K_{3,3}$ be a planar graph. Then, by above theorem, $\epsilon \leq 2n-4$. In $K_{3,3}$, we have $n = 6$ and $\epsilon = 9$. Hence $2n-4 = 8 < \epsilon = 9$, which contradicts the previous result. Hence, $K_{3,3}$ is non-planar. □

The *girth* of a graph is the length of its smallest cycle. Then, the following theorem is a generalisation of Theorem 6.5.2.

Theorem 6.5.6. *Let G be a plane graph with n vertices and ϵ edges and let g be the girth of the graph G. Then, $\epsilon \leq \frac{g(n-2)}{g-2}$.*

Proof. Let f be the number of faces of G and let m_i be the number of edges in the boundary of the i-th face, where $i = 1, 2, \ldots, f$. Since g is the girth of G every face contains at least g edges, we have $m_i \geq g$ for all $i = 1, 2, 3, \ldots, f$. Then,

$$gf \leq \sum_{i=1}^{f} m_i. \tag{6.5}$$

Since every edge can be in the boundary of at most two faces, we have

$$\sum_{i=1}^{f} m_i \leq 2\epsilon. \tag{6.6}$$

From equations (6.5) and (6.6), we have $gf \leq 2\epsilon$ or $f \leq \frac{2}{g}\epsilon$. Now, by Euler's theorem, we have

$$\begin{aligned} n+f-\epsilon &= 2 \\ \implies n+\frac{2}{g}\epsilon-\epsilon &\leq 2 \\ \implies n-\frac{(g-2)\epsilon}{g} &\geq 2 \\ \implies \frac{(g-2)\epsilon}{g} &\leq n-2 \\ \implies \epsilon &\leq \frac{g(n-2)}{g-2}. \end{aligned}$$

This completes the proof. □

Let ϕ be a region of a planar graph G . We define the degree of ϕ, denoted by $d(\phi)$, as the number of edges on the boundary of ϕ.

Theorem 6.5.7. *Let G be a plane graph. Then,* $\sum_{f \in F(G)} d(f) = 2|E(G)|$.

A graph is *d-degenerate* if it has no subgraph of minimum degree at least d.

Problem 6.5.8. Using Euler Theorem on Planar Graphs, verify whether the Petersen's graph is planar.

Solution. Petersen graph does not have triangles or 4-cycles. The smallest cycle in Petersen graph is C_5. Therefore, its girth $g = 5$. Then, if Petersen graph were planar, then by Theorem 6.5.6, it should satisfy the inequality $\epsilon \leq \frac{g(n-2)}{g-2}$.

For Petersen graph, we have $n = 10, \epsilon = 15$ and $g = 5$. Then, $\epsilon \leq \frac{g(n-2)}{g-2} \implies 15 \leq \frac{5\times 8}{3}$, which is not true. Therefore, Petersen graph is not planar. □

Problem 6.5.9. Using Euler Theorem on Planar Graphs, verify whether the Dürer graph (see Figure 6.11) is planar.

Solution: From Figure 6.11, we note that the Dürer graph G has triangles. Therefore, if G is planar, then it should satisfy the inequality $\epsilon \leq 3n-6$.

For the Dürer graph, we have $n = 12, \epsilon = 18$ and $g = 3$. Therefore, $\epsilon \leq 3n-6 \implies 18 \leq 3\times 12-6$, which is true. Therefore, the Dürer graph is planar. (Note that

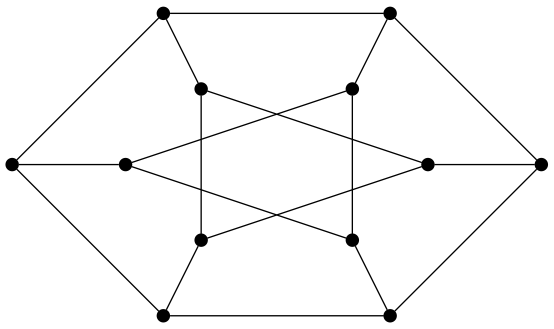


Figure 6.11: Dürer graph

one of the two triangles inside the outer-cycle of the Dürer graph can be redrawn to the outside of the cycle so that no two edges of the new graph do not intersect.)

6.6 Geometric Dual of a Graph

Definition 6.6.1 (Geometric Dual of a Graph)**.** Given a plane graph G, the *dual graph* (usually called the *geometric dual*) of G, denoted by G^*, is the plane graph whose vertices are the faces of G such that two vertices v_i^* and v_j^* in G^* are adjacent in G^* if and only if the corresponding faces f_i and f_j are adjacent in G.

Note that two faces of a graph G are adjacent in G if they have a common edge at their boundaries.

In other words, the correspondence between edges of G and those of the dual G^* is as follows:

If $e \in E(G)$ lies on the boundaries of two faces f_i and f_j in G, then the endpts of the corresponding dual edge $e^* \in E(G^*)$ are the vertices v_i^* and v_j^* that represent faces f_i and f_j of G.

A graph and its dual are illustrated in Figure 6.12.

Remark 6.6.2. Note that there will be a one-to-one correspondence between the edges of a graph G and its dual G^*-one edge of G^* intersecting one edge of G (see Figure6.12). We note that

$$\begin{aligned} |V(G^*)| &= |F(G)| \\ |E(G^*)| &= |E(G)| \\ |F(G^*)| &= |V(G)| \end{aligned}$$

Also, if r and μ respectively denote the rank and nullity of G and r^* and μ^* denote the rank and nullity of G^*, then we observe that $r = \mu^*$ and $\mu = r^*$.

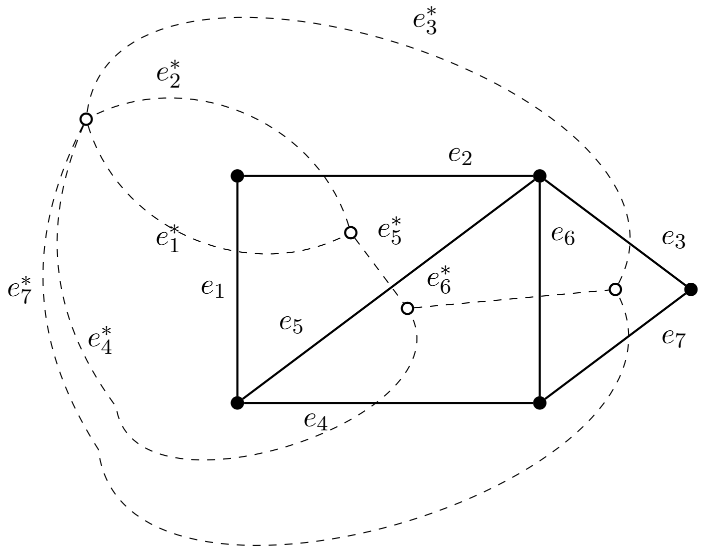


Figure 6.12: A graph and its dual graph

The following observations can be made on the relationships between a planar graph G and its dual G^*:

(i) A self-loop in G corresponds to a pendant edge in G^*;
(ii) A pendant edge in G corresponds to a self-loop in G^*;
(iii) Edges that are in series in G produce parallel edges in G^*;
(iv) Parallel edges i G produce edges in series in G^*;
(v) The number of edges on the boundary of a face f (the degree of f) in G is equal to the degree of the corresponding vertex v^* in G^*.
(vi) Both G and G^* are planar.
(vii) $G^{**} = G$. That is, G is the dual of G^*.

It can be observed that duals of isomorphic graphs need not be isomorphic. The Figure 6.13 illustrates this fact.

From this fact, we notice that *the two different geometric dual graphs of the same graph (isomorphic) need not be isomorphic.*

Theorem 6.6.3. *Two planar graphs G_1 and G_2 are duals of each other if and only if there exists a one-to-one correspondence between their edge sets such that the cycles (cycles) in G_1 corresponds to cut-sets in G_2 and vice versa.*

Proof. Let us consider a plane representation of a planar graph. Let us also draw a dual G^* of G. Then consider an arbitrary cycle C in G.

Clearly, C will form some closed simple curve in the plane representation of G - dividing the plane into two areas.(Jordan curve theorem). Thus the vertices of

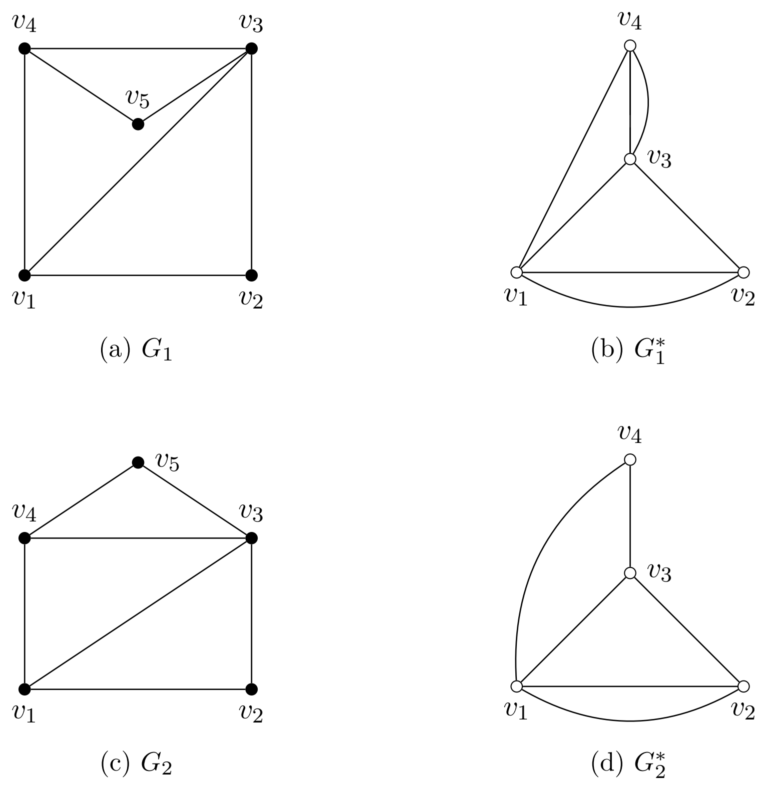


Figure 6.13: Two isomorphic graphs and their non-isomorphic duals

G^* are partitioned into two nonempty, mutually exclusive subsets - one inside C and the other outside. In other words, the set of edges C^* in G^* corresponding to the set C in G is a cut-set in G^*.(No proper subset of C^* will be a cut-set in G^*). Likewise, it is apparent that corresponding to a cut-set S^* in G^* there is a unique cycle consisting of the corresponding edge-set S in G such that S is a cycle. This proves the necessary part of the theorem.

To prove the sufficiency, let G a planar graph let G' be a graph for which there is a one-to-one correspondence between the cut-set of G and cycles of G', and vice versa. Let $G*$ be a dual graph of G. There is a one-to-one correspondence between the cycles of G' and cut-sets of G, and also between the cut-sets of G and cycles of G^*. Therefore there is a one-to-one correspondence between the cycles of G' and G^*, implying that G' and G^* are 2-isomorphic. Hence, G' must be a dual of G. □

6.6.1 Dual of a Subgraph

Let G be a planar graph and G^* be its dual. Let e be an edge in G, and the corresponding edge in G^* be e^*. Suppose that we delete edge e from G and then try to find the dual of $G-e$. If edge e was on the boundary of two regions, removal of e would merge these two regions into one. Thus, the dual $(G-e)^*$ can be obtained from G^* by deleting the corresponding edge e^* and then fusing the two end vertices of a^* in $G^* - e^*$. On the other hand, if edge e is not on the boundary, e^* forms a self-loop. In that case, $G^* - e^*$ is the same as $(G-e)^*$. Thus, if a graph G has a dual G^*, the dual of any subgraph of G can be obtained by successive application of this procedure.

6.6.2 Dual of a Homeomorphic Graph

Let G be a planar and G^* be its dual. Let a be an edge in G, and the corresponding edge in G^* be a^*. Suppose that we create an additional vertex in G by introducing a vertex of degree two in edge a. It simply adds an edge parallel to a^* in G^*. Likewise, the reverse process of merging two edges in series will simply eliminate one of the corresponding parallel edges in G^*. Thus if a graph G has a dual G^*, the dual of any graph homeomorphic to G can be obtained from G^* by the above procedure.

Theorem 6.6.4. *A graph has a dual if and only if it is planar.*

Proof. The necessary part is obvious. We can draw the dual of a graph G only when we can identify each face of G, which is possible only when G is planar.

Conversely, assume that G has a dual. Now, we have to prove that G is planar. Let G be a non-planar graph. Then, according to Kuratowski's theorem, G contains K_5 or $K_{3,3}$ or a graph homeomorphic to either of these. We have already seen that a graph G can have a dual if every subgraph H of G and every graph homeomorphic to H has a dual. Thus, if we can show that neither K_5 nor $K_{3,3}$ has a dual, our result will be completed. Now, we have the following cases.

1. Suppose that $K_{3,3}$ has a dual D. Observe that the cut-sets in $K_{3,3}$ correspond to cycles in D and vice versa. Since $K_{3,3}$ has no cut-set consisting of two edges (as at least three edges are to be removed form $K_{3,3}$ to make it disconnected), D has no cycle consisting of two edges. That is, D contains no pair of parallel edges. Since every cycle in $K_{3,3}$ is of length four or six, D has no cut-set with less than four edges. Therefore, the degree of every vertex in D is at least four. As D has no parallel edges and the degree of every vertex is a least four, D must have at least five vertices each of degree four or more. That is, D must have at least $(5 \times 4)/2 = 10$ edges (since, by the first theorem on graph theory, we have $\epsilon = \frac{1}{2}\sum d(v)$). This is a contradiction to the fact that $K_{3,3}$ and its dual has nine edges. Hence, $K_{3,3}$ cannot have a dual.
2. Suppose that the graph K_5 has a dual H. We know that K_5 has
   (a) 10 edges, none of which are parallel edges;
   (b) no cut-set with two edges (as at least four edges must be removed to make K_5 disconnected); and
   (c) cut-sets with only four or six edges.

   Consequently, graph H must have
   (a) 10 edges,

(b) no vertex with degree less than three,
(c) no pair of parallel edges, and
(d) cycles of length four or six only.

Now, graph H contains a hexagon, and no more than three edges can be added to a hexagon without creating a cycle of length three or a pair of parallel edges. Since both of these are forbidden in H and H has 10 edges, there must be at least 7 vertices in H. The degree of each of these vertices is at least three. This leads to H having at least 11 edges, which is a contradiction. Hence, K_5 also does not have a dual.

This completes the proof. □

Definition 6.6.5 (Self-dual Graphs)**.** A graph G is said to be a *self-dual graph* if it is isomorphic to its geometric dual G^*.

For example, the complete graph K_4 is a self-dual graph. But the dual of complete graph K_3 has only two faces and hence it is not a self-dual graph. For $n \geq 5$, the complete graph K_n is not planar and hence its dual cannot be constructed. Thus, K_4 is only non-trivial complete graph which is a self-dual graph.

Problem 6.6.6. Draw the geometrical dual G^* of the graph G given below and also check whether G and G^* are self-duals or not. Substantiate your answer clearly.

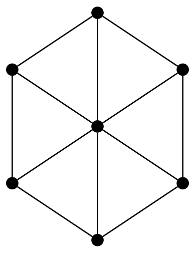


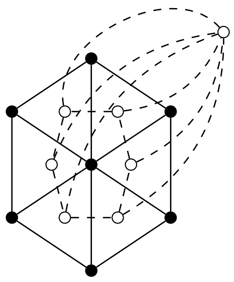
Figure 6.14: G

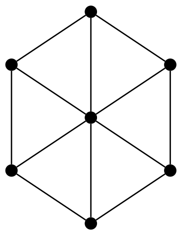
Solution: The dual of G can be drawn as shown in Figure 6.15:

From the graphs G and G^* drawn above, it can be noted that both graphs are isomorphic to each other. Therefore, G and G^* are self-duals. □

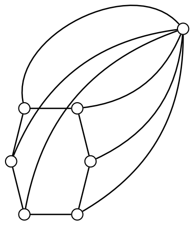
Problem 6.6.7. Is a graph with degree sequence $(4,3,3,3,2,1)$ planar? If so, find the number of faces in G.

Solution: Here, the number of vertices in G is 6. That is $n = 6$. Now, by First Theorem of Graph Theory, $\epsilon = \frac{1}{2}\sum\limits_{v \in V} d(v)$. Therefore, $\epsilon(G) = \frac{1}{2}(4+3+3+3+2+1) = 8$. By a result, we have for any connected planar graph, $\epsilon \leq 3n-6$. That is, $8 \leq 3 \times 6 - 6 = 12$, which is true. Therefore, the corresponding graph is planar. □





(a) A graph G.



(b) The dual G^* of G

Figure 6.15: A graph G and its dual G^*

Problem 6.6.8. An r-regular graph of order 12 is embedded on a plane, resulting in 8 faces. Then find the value of r.

Solution: Given that $n = 12$ and $f = 8$. By Euler theorem on plane graphs, we have

$$\begin{aligned} n + f - \epsilon &= 2 \\ 12 + 8 - \epsilon &= 2 \end{aligned}$$

Therefore, $\epsilon = 18$. Since the graph is r-regular, we have $\epsilon = \frac{rn}{2} = 6r$. That is, $6r = 18$ and $r = 3$. □

Theorem 6.6.9. *Every subgraph of a planar graph is planar.*

Proof. If G is planar, then there exists a planar embedding, say G' of G. For every subgraph H of G, we can find the vertices and edges of H in the planar embedding of G. The graph obtained from G' by removing its vertices and/or edges, which are not in H, is a planar embedding of H. Therefre, H is also planar. □

Theorem 6.6.10. *A bipartite graph with $\delta(G) \geq 4$ is non-planar.*

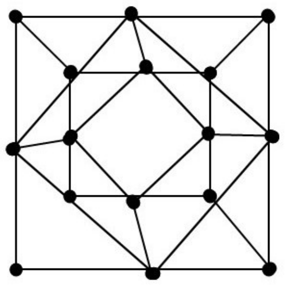
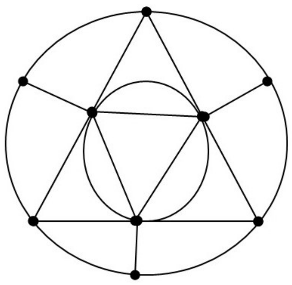
Proof. Let G be a bipartite graph with $\delta(G) \geq 4$. Since every vertex of G possesses degree 4 or more, we can find out a subgraph of G which is isomorphic to $K_{3,3}$. Therefore, Kuratowski's theorem, G is not planar. □

6.7 Exercise Problems

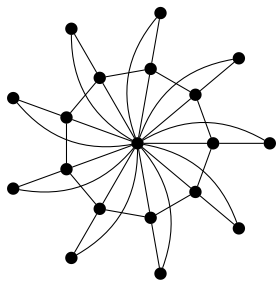
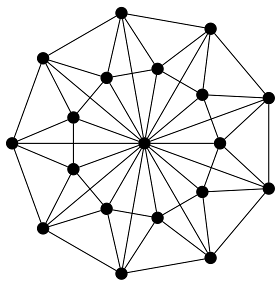
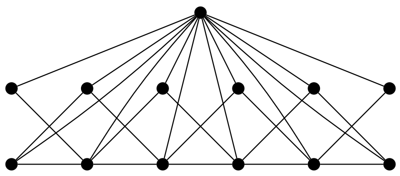
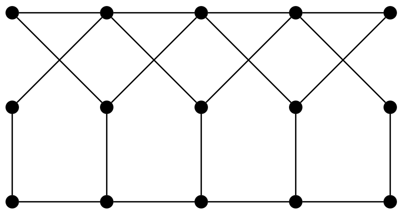
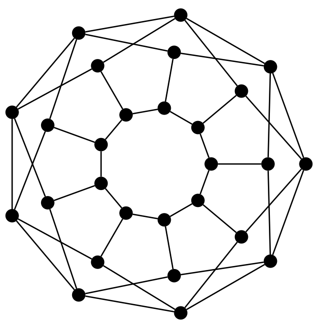
1. Show that Petersen graph is 3-connected.
2. Show that $\lambda(G) = \kappa(G)$ if G is a simple graph with $\Delta(G) \leq 3$.
3. Show that deleting an edge-cut of size 3 from Petersen graph creates an isolated vertex.
4. Show that deletion of an edge reduces the connectivity by at most 1.
5. Show that a bipartite graph G with $d(G) \geq 4$ is non-planar.
6. Prove that plane graph G is bipartite if and only if every face of G has even length (or degree). [*Hint*: Use induction on f]
7. Prove that the complement of a simple planar graph with at least 11 vertices is non-planar.
8. Let G be a connected graph with 15 vertices and 40 edges. Can we say that G is planar? Why?
9. Let G be a connected triangle-free non-bipartite graph with 10 vertices and 25 edges. Can we say that G is planar? Why?
10. Draw a planar graph on 10 vertices and find if dual.
11. If e is an edge of the complete graph K_5, then show that $K_5 - e$ is planar.
12. If e is an edge of the complete bipartite graph $K_{3,3}$, then show that $K_{3,3} - e$ is planar.
13. Prove that a subset of a planar graph is also planar.
14. Let G be a 4-regular graph with 10 faces. Determine its order. Give a drawing of such a graph.
15. If G is a graph order 11, then show that either G or its complement $\overline{G}$ is non-planar.
16. Show that a simple planar graph G with fewer than 12 vertices, has a vertex v with degree $d(v) \leq 4$.
17. Show that a simple planar graph G with fewer than 30 edges, has a vertex v with degree $d(v) \leq 4$.
18. Let G be a plane graph with n vertices, ϵ edges, k components and f faces. Then, show that $n - \epsilon + f = k + 1$.
19. Let G be a connected plane graph. Then, show that G is bipartite if and only if its dual G^* is an Eulerian graph.
20. Find the number of faces in planar graph whose degree sequence is $(5,4,4,3,3,3,2,2,2)$. [Hint: Use handshaking lemma and Euler Theorem on plane graphs]
21. Is the graph with degree sequence $(4,3,3,3,2,1)$ planar? If so, find its number of regions.
22. Is it possible to draw a plannar graph with 10 vertices and 25 vertices? Justify your answer.
23. Is it possible to draw a bipartite planar graph whose order 10 vertices and 18 edges? Justify your answer.
24. Draw an example of a simple graph in which the degree of every vertex is at least 5.
25. Let G be a connected simple plane graph. Then show that

(a) if $d(v) \geq 5$, for all vertex v in $V(G)$, then there are at least 12 vertices of degree 5 in G.
(b) if $n \geq 4$ and $d(v) \geq 3$, for all vertex v in $V(G)$, then G has at least 4 vertices of degree 6.

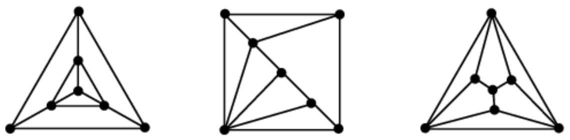
26. Verify Euler's theorem for the following graphs:

27. Determine whether the following graphs are planar. Justify your answer.

28. Find the intersection number of the graphs given in problem 27.



29. Check whether the following graphs are self-duals or not:

****

Chapter 7

Matrix Representations of Graphs

****

Matrices are an alternate way to represent and summarize network data. A matrix contains exactly the same information as a graph, but is more useful for computation and computer analysis. Indeed, with a given graph, adequately labelled, there are associated several matrices.

7.1 Incidence Matrix of a Graph

Let G be a graph with n vertices, m edges and without self-loops. The *incidence matrix* A of G is an $n \times m$ matrix defined by $A(G) = [a_{i,j}]; 1 \leq i \leq n,\ 1 \leq j \leq \epsilon$, where

$$a_{ij} = \begin{cases} 1; & \text{if } j\text{-th edge incidents on } i\text{-th vertex} \\ 0; & \text{Otherwise.} \end{cases}$$

where the n rows of A of G correspond to the n vertices and the m columns of A correspond to ϵ edges.

The incidence matrix contains only two types of elements, 0 and 1. Hence, this is clearly a *binary matrix* or a $(0, 1)$-matrix.

The following table gives the incidence relation of the graph G:

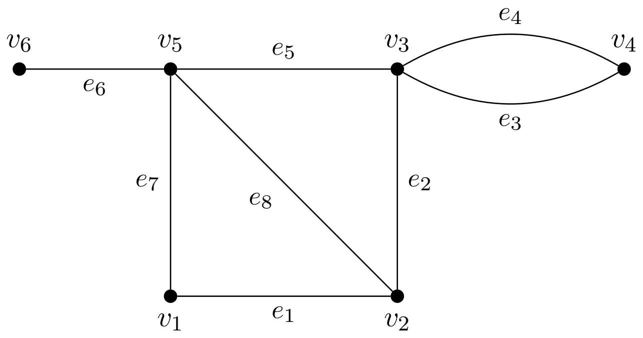


Figure 7.1: A graph G

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_1	1	0	0	0	0	0	1	0
v_2	1	1	0	0	0	0	0	1
v_3	0	1	1	1	1	0	0	0
v_4	0	0	1	1	0	0	0	0
v_5	0	0	0	0	1	1	1	1
v_6	0	0	0	0	0	1	0	0

Table 7.1: The incidence relation of the graph G

.

Therefore, the incidence matrix of G is as given below:

$$A(G) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Now consider the following disconnected graph G with two components.

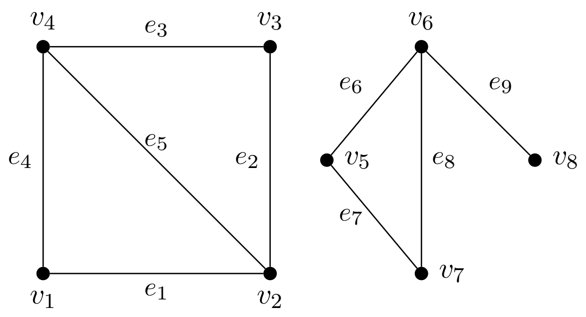


Figure 7.2: A disconnected graph G with two components G_1 and G_2.

The incidence matrix of the graph in Figure 7.2 is given below:

$$A(G) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

From the above examples, we have the following observations about the incidence matrix A of a graph G.

1. Since every edge is incident on exactly two vertices, each column of A has exactly two ones.
2. The number of ones in each row equals the degree of the corresponding vertex.
3. A row with all zeros represents an isolated vertex.
4. Parallel edges in a graph produce identical columns in its incidence matrix.
5. If a graph G is disconnected and consists of two components G_1 and G_2, then its incidence matrix $A(G)$ can be written in a block diagonal form as

$$A(G) = \begin{bmatrix} A(G_1) & 0 \\ 0 & A(G_2) \end{bmatrix}$$

   where $A(G_1)$ and $A(G_2)$ are the incidence matrices of the components G_1 and G_2 of G. This observation results from the fact that no edge in G_1 is incident on vertices of G_2 and vice versa. Obviously, this is also true for a disconnected graph with any number of components.
6. Permutation of any two rows or columns in an incidence matrix simply corresponds to relabelling the vertices and edges of the same graph.

Remark 7.1.1. *The matrix A has been defined over a field, Galois field modulo* 2 *or* $GF(2)$ *(that is, the set* $0, 1$ *with operation addition modulo* 2 *written as* $+$ *such that* $0+0=0$, $1+0=1$, $1+1=0$ *and multiplication modulo* 2 *written as"." such that* $0 \cdot 0 = 0$, $1.0 = 0 = 0.1$, $1.1 = 1$.

Theorem 7.1.2. *Two graphs* G_1 *and* G_2 *are isomorphic if and only if their incidence matrices* $A(G_1)$ *and* $A(G_2)$ *differ only by permutation of rows and columns.*

Proof. Let the graphs G_1 and G_2 be isomorphic. Then, there is a one-one correspondence between the vertices and edges in G_1 and G_2 such that the incidence relation is preserved. Hence, $A(G_1)$ and $A(G_2)$ are either same or differ only by permutation of rows and columns.

The converse follows, since permutation of any two rows or columns in an incidence matrix simply corresponds to relabelling the vertices and edges of the same graph. This complete the proof. $\square$

7.1.1 Rank of the Incidence Matrix

Let G be a graph and let $A(G)$ be its incidence matrix. Now, each row in $A(G)$ is a vector over $GF(2)$ in the vector space of graph G. Let the row vectors be denoted by $A_1, A_2, \dots, A_n$. Then,

$$A(G) = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n. \end{bmatrix}$$

Since there are exactly two 1's in every column of A, the sum of all these vectors is 0 (this being a modulo 2 sum of the corresponding entries). Thus vectors $A_1, A_2, \dots, A_n$ are linearly dependent. Therefore, $rank A < n$. Hence, $rank A \le n-1$.

From the above observations, we have the following result.

Theorem 7.1.3. *If $A(G)$ is an incidence matrix of a connected graph G with n vertices, then rank of $A(G)$ is $n-1$.*

Proof. Let G be a connected graph with n vertices and let the number of edges in G be ϵ. Let $A(G)$ be the incidence matrix and let $A_1, A_2, \dots, A_n$ be the row vector of $A(G)$. Then, we have

$$A(G) = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n. \end{bmatrix} \tag{7.1}$$

Since $A_1 + A_2 + \dots + A_n = \begin{bmatrix} 2 & 2 & \dots & 2 \end{bmatrix} \equiv 0(\text{mod } 2)$, we have $A_1, A_2, \dots A_n$ are linearly dependent. Hence, $rank\,(A(G)) \le n-1$.

Consider the sum of any m of these row vectors, $\epsilon \le n-1$. Since G is connected, $A(G)$ cannot be partitioned in the form

$$A(G) = \begin{bmatrix} A(G_1) & 0 \\ 0 & A(G_2) \end{bmatrix} \tag{7.2}$$

such that $A(G_1)$ has ϵ rows and $A(G_2)$ has $n-\epsilon$ rows.

Thus, there exists no $m \times m$ submatrix of $A(G)$ for $m \le n-1$, such that the modulo 2 sum of these m rows is equal to zero. As there are only two elements 0 and 1 in this field, the additions of all vectors taken m at a time for $m = 1, 2, \dots, n-1$ gives all possible linear combinations of $n-1$ row vectors. Hence, no linear combinations of m row vectors of A, for $m \leq n-1$, is zero. Therefore, $rank(A(G)) \geq n-1$.

Combining Equation (7.1) and Equation (7.2), it follows that $rank(A(G)) = n-1$. □

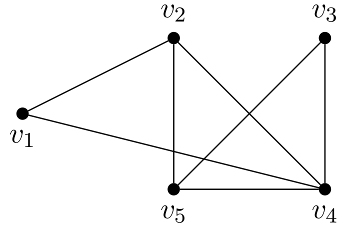
Corollary 7.1.4. *If G is a disconnected graph with k components, then $rank(A(G)) = n-k$.*

Let G be a connected graph with n vertices and m edges. Then, the order of the incidence matrix $A(G)$ is $n \times \epsilon$. Now, if we remove any one row from $A(G)$, the remaining $(n-1)\times\epsilon$ submatrix is of rank $(n-1)$. Hence, the remaining $(n-1)$ row vectors are linearly independent, which shows that only $(n-1)$ rows of an incidence matrix are required to specify the corresponding graph completely, because $(n-1)$ rows contain the same information as the entire matrix. This follows from the fact that given $(n-1)$ rows, we can construct the n-th row, as each column in the matrix has exactly two ones. Such an $(n-1)\times\epsilon$ matrix of A is called a *reduced incidence matrix* and is denoted by A_f. The vertex corresponding to the deleted row in A_f is called the *reference vertex*. Obviously, any vertex of a connected graph can be treated as the reference vertex.
The following result gives the nature of the incidence matrix of a tree:

Problem 7.1.5. Draw the graph with the following matrix as its incidence matrix.

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

Solution: From the matrix, we note that the graph required has five vertices, say v_1, v_2, v_3, v_4 and v_5 and seven edges say $e_1, e_2, e_3, e_4, e_5, e_6$ and e_6. Therefore, the graph corresponding to the given matrix is given below:



□

7.2 Cycle Matrix

Definition 7.2.1. Let G be a graph with ϵ edges and q different cycles. The *cycle matrix* or *circuit matrix* of G, denoted by $B(G)$, is defined as a $(0,1)$-matrix $B(G) = [b_{ij}]$ of order $q \times \epsilon$, such that

$$b_{ij} = \begin{cases} 1; & \text{if the } i\text{-th cycle includes } j\text{-th edge;} \\ 0; & \text{otherwise.} \end{cases} \tag{7.3}$$

In Figure 7.1, we have the the following cycles:

$$\begin{aligned} C_1 &= \{e_1, e_7, e_8\} \\ C_2 &= \{e_2, e_5, e_8\} \\ C_3 &= \{e_3, e_4\} \\ C_4 &= \{e_1, e_2, e_5, e_7\}. \end{aligned}$$

Therefore, the cycle matrix of the given graph in Figure 7.1 is a 4×9 matrix as given below:

$$C(G) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Now consider the following disconnected graph:

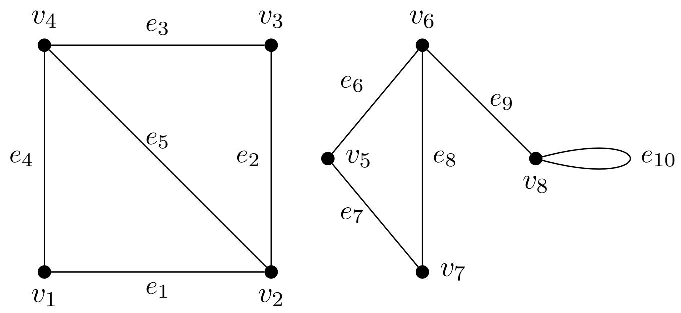


Figure 7.3: A disconnected graph G with two components G_1 and G_2.

In Figure 7.3, we have the the following cycles:

$$\begin{aligned} C_1 &= \{e_1, e_4, e_5\} \\ C_2 &= \{e_2, e_3, e_5\} \\ C_3 &= \{e_1, e_2, e_3, e_4\} \\ C_4 &= \{e_6, e_7, e_8\} \end{aligned}$$

$$C_5 = \{e_{10}\}.$$

Therefore, the cycle matrix of the given graph in Figure 7.3 is a 5×10 matrix as given below:

$$B(G) = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In view of the above examples, we have the following observations regarding the cycle matrix $B(G)$ of a graph G:

1. A column of all zeros corresponds to a cut-edge. That is, an edge which does not belong to any cycle corresponds to an all-zero column in $B(G)$.
2. Each row of $B(G)$ is a cycle vector.
3. A cycle matrix has the property of representing a self-loop and the corresponding row has a single 1.
4. The number of 1's in a row is equal to the number of edges in the corresponding cycle.
5. If the graph G is separable (or disconnected) and consists of two blocks (or compo- nents) H_1 and H_2, then the cycle matrix $B(G)$ can be written in a block-diagonal form as

$$B(G) = \begin{bmatrix} B(H_1) & 0 \\ 0 & B(H_2) \end{bmatrix} \tag{7.4}$$

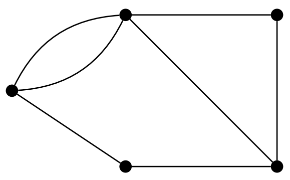
   where $B(H_1)$ and $B(H_2)$ are the cycle matrices of H_1 and H_2. This is obvious from the fact that cycles in H_1 have no edges belonging to H_1 and vice versa.
6. Permutation of any two rows or columns in a cycle matrix corresponds to relabeling the cycles and the edges.
7. Two graphs G_1 and G_2 have the same cycle matrix if and only if G_1 and G_2 are 2-isomorphic as two graphs G_1 and G_2 are 2-isomorphic if and only if they have cycle correspondence. Hence, we observe that the cycle matrix does not specify a graph completely, but only specifies the graph within 2-isomorphism.

The following two graphs which are non-isomorphic, but has the same cycle matrix.

The cycle matrix of both matrices will be as follows:

$$B(G) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

This fact highlights the fact that the cycle matrix does not specify a graph in


(a) G

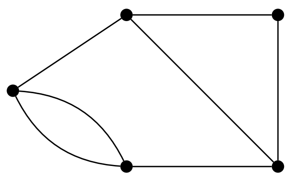

(b) H

Figure 7.4: Examples non-isomorphic graphs which have same cycle matrix

the full sense.

7.3 Cut-Set Matrix

Let G be a graph with ϵ edges and q cut-sets. The *cut-set matrix* of G is a matrix, denoted by $C(G)$ and is defined to be $C = [c_{ij}]_{q\times\epsilon}$ of G is a $(0,1)$-matrix such that

$$c_{ij} = \begin{cases} 1; & \text{if } i\text{-th cutset contains } j\text{-th edge;} \\ 0; & \text{otherwise.} \end{cases}$$

The cut-sets of the graph in Figure 7.1 are $c_1 = \{e_1, e_7\}, c_2 = \{e_1, e_2, e_7\}, c_3 = \{e_3, e_4\}, c_4 = \{e_2, e_5\}, c_5 = \{e_5, e_7, e_8\}, c_6 = \{e_6\}, c_7 = \{e_2, e_7, e_8\}$ and $c_8 = \{e_1, e_5, e_8\}$.

The following table gives the corresponding relation between the cutsets and edges of the graph.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
c_1	1	0	0	0	0	0	1	0
c_2	1	1	0	0	0	0	0	1
c_3	0	0	1	1	0	0	0	0
c_4	0	1	0	0	1	0	0	0
c_5	0	0	0	0	1	0	1	1
c_6	0	0	0	0	0	1	0	0
c_7	0	1	0	0	0	0	1	1
c_8	1	0	0	0	1	0	0	1

Table 7.2: Table for cut-set matrix of a graph.

Hence the cut-set matrix of G is

$$C(G) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

The following are some important observations about the cut-set matrix $C(G)$ of a graph G.

1. The permutation of rows or columns in a cut-set matrix corresponds simply to the renaming of the cut-sets and edges respectively.
2. Each row in $C(G)$ is a cut-set vector.
3. A column with all zeros corresponds to an edge forming a self-loop.
4. Parallel edges form identical columns in the cut-set matrix.
5. In a non-separable graph, since every set of edges incident on a vertex is a cut-set, every row of incidence matrix $A(G)$ is included as a row in the cut-set matrix $C(G)$. That is, for a non-separable graph G, $C(G)$ contains $A(G)$. For a separable graph, the incidence matrix of each block is contained in the cut-set matrix.
   For example, the incidence matrix of the block $\{e_1, e_2, e_5, e_7, e_8\}$ of the graph G in Figure 7.1is the 4×5 submatrix of C,

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

   which is obtained after after removing the rows c_3, c_6, c_7, c_8 and columns e_3, e_4, e_6.
6. It follows from observation 5 , that $rank(C(G)) \geq rank(A(G))$. Hence, for a connected graph with n vertices, $rank(C(G)) \geq n-1$.

7.4 Adjacency Matrix

As an alternative to the incidence matrix, it is sometimes more convenient to represent a graph by its adjacency matrix or connection matrix. The *adjacency matrix* of a graph G with n vertices and no parallel edges is an n by n symmetric binary matrix $X = [x_{ij}]$ defined over the ring of integers such that

$$x_{ij} = \begin{cases} 1; & \text{if there is an edge between } i\text{-th and } j\text{-th vertices, and} \\ 0; & \text{if there is no edge between them.} \end{cases}$$

Consider the following graph G without parallel edges and self-loops.

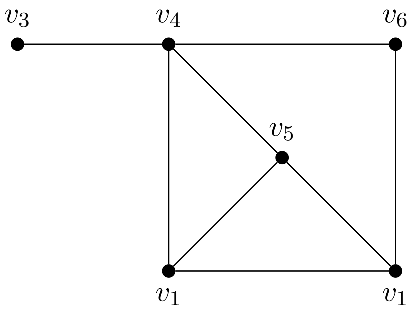


Figure 7.5: A graph G

The adjacency matrix of the graph G in Figure 7.5 is given by

$$X(G) = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The adjacency matrix of the disconnected graph given in Figure 7.2 is as follows:

$$X(G) = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

By the definition of the adjacency matrix of a graph G, we can notice that the adjacency matrices can be defined for the graphs having self-loops also. For example, consider the following graph and its adjacency matrix.

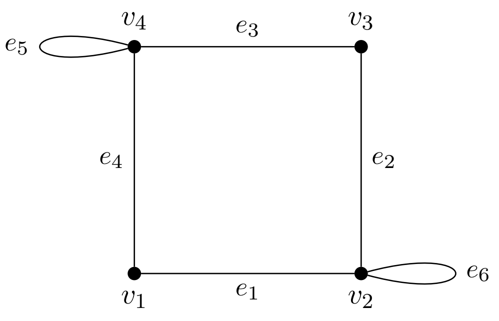


Figure 7.6: A graph G with self-loops

$$X(G) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

Remark 7.4.1. Observations that can be made immediately about the adjacency matrix X of a graph G are:

1. The entries along the principal diagonal of X are all $0'$s if and only if the graph has no self-loops. A self-loop at the ith vertex corresponds to $x_{ii} = 1$.
2. The definition of adjacency matrix makes no provision for parallel edges. This is why the adjacency matrix X was defined for graphs without parallel edges.
3. If the graph has no self-loops (and no parallel edges), the degree of a vertex equals the number of $1'$s in the corresponding row or column of X.
4. Permutations of rows and of the corresponding columns imply reordering the vertices. It must be noted, however, that the rows and columns must be arranged in the same order. Thus, if two rows are interchanged. Hence, two graphs G_1 and G_2 with no parallel edges are isomorphic if and only if their adjacency matrices $X(G_1)$ and $X(G_2)$ are related:

$$X(G_2) = R^{-1} \cdot X(G_1) \cdot R,$$

   where R is a permutation matrix.
5. A graph G is disconnected and is in two components G_1 and G_2 if and only if its adjacency matrix $X(G)$ can be partitioned as

$$X(G) = \begin{bmatrix} X(G_1) & O \\ 0 & X(G_2) \end{bmatrix},$$

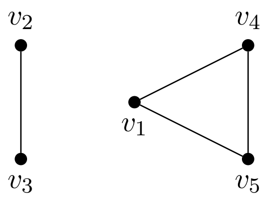
   where $X(G_1)$ is the adjacency matrix of the component G_1 and $X(G_2)$ is that of the component G_2. This partitioning clearly implies that there exists no edge joining any vertex in subgraph G_1 to any vertex in subgraph G_2.
6. Given any square, symmetric, binary matrix Q of order n, one can always

construct a graph G of n vertices (and no parallel) such that Q is the adjacency matrix of G.

Problem 7.4.2. Draw the graph, with the following matrix as its sdjacency matrix:

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

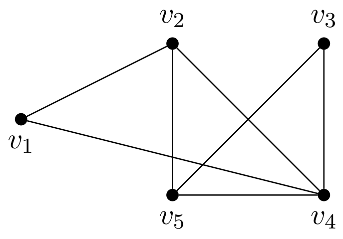
Solution: From the matrix, we note that the graph required has five vertices, say v_1, v_2, v_3, v_4 and v_5. Also, we have the adjacency relations: $v_1 \sim v_4, v_1 \sim v_5, v_2 \sim v_3, v_3 \sim v_2, v_4 \sim v_1, v_4 \sim v_5, v_5 \sim v_1$ and $v_5 \sim v_4$. Therefore, the graph corresponding to the given matrix is given below:



Problem 7.4.3. Draw the graph, with the following matrix as its sdjacency matrix:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 10 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Solution: From the matrix, we note that the graph required has five vertices, say v_1, v_2, v_3, v_4 and v_5. Also, we have the adjacency relations: $v_1 \sim v_2, v_1 \sim v_4, v_2 \sim v_3, v_3 \sim v_2, v_4 \sim v_1, v_4 \sim v_5, v_5 \sim v_1$ and $v_5 \sim v_4$. Therefore, the graph corresponding to the given matrix is given below:



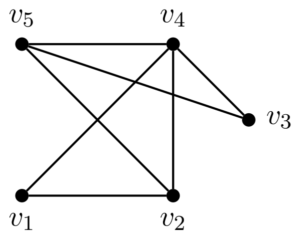
Problem 7.4.4. Draw the graph represented by the following adjacency matrix:

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

Solution: The above matrix can be expressed in tabular form as follows:

	v_1	v_2	v_3	v_4	v_5
v_1	0	1	0	1	0
v_2	1	0	0	1	1
v_3	0	0	0	1	1
v_4	1	1	1	0	1
v_5	0	1	1	1	0

Hence, the graph required is given below:



□

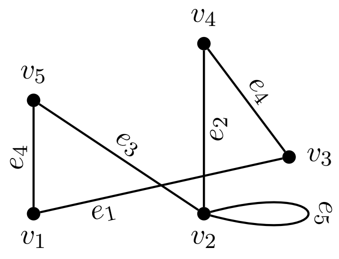
Problem 7.4.5. Draw the graph represented by the following adjacency matrix:

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

Solution: The above matrix can be expressed in tabular form as follows:

	e_1	e_2	e_3	e_4	e_5	e_6
v_1	1	0	0	0	0	1
v_2	0	1	1	0	1	0
v_3	1	0	0	1	0	0
v_4	0	1	0	1	0	0
v_5	0	0	1	0	0	1

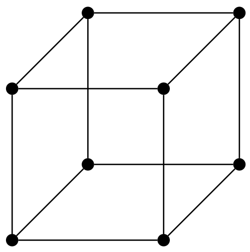
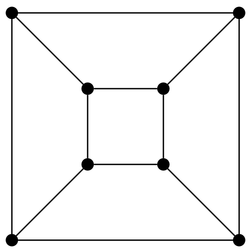
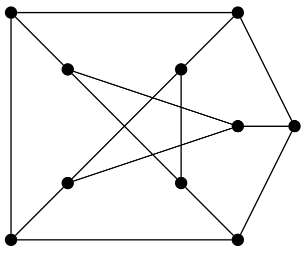
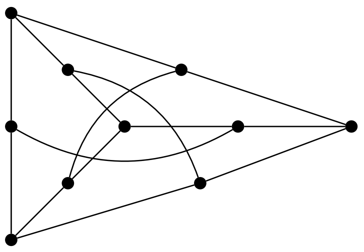
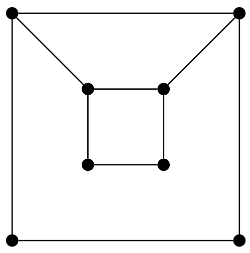
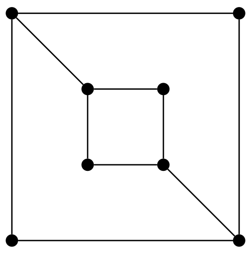
Hence, the graph required is given below:



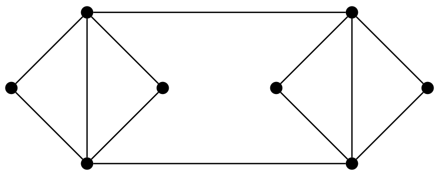
□

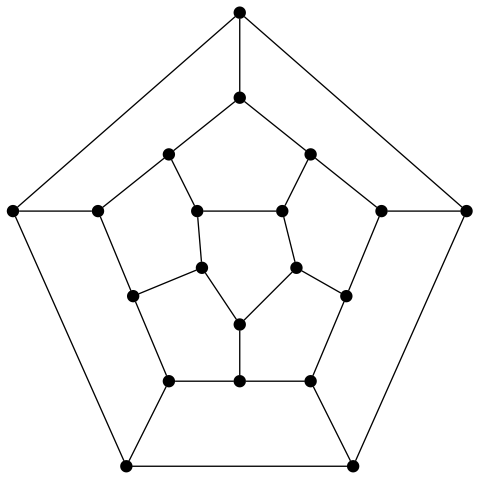
7.5 Exercise Problems

1. Find the incidence matrix, cycle matrix, cut-set matrix and adjacency matrix for the following graphs:

2. Find the graphs whose adjacency matrices are





(a) $$\begin{bmatrix} 0&1&1&0&0&0&0\\ 1&0&1&1&0&1&0\\ 1&1&0&1&0&0&1\\ 0&1&1&0&0&0&0\\ 0&0&0&0&0&1&1\\ 1&0&0&0&1&0&1\\ 0&0&0&0&1&1&0 \end{bmatrix};$$

(b) $$\begin{bmatrix} 0&1&1&1&1\\ 1&0&1&1&1\\ 1&1&0&1&0\\ 1&1&1&0&1\\ 1&1&0&1&0 \end{bmatrix}$$

(c) $$\begin{bmatrix} 0&1&0&1&1&0&0&0\\ 1&0&1&0&0&1&0&0\\ 0&1&0&1&0&0&1&0\\ 1&0&1&0&0&0&0&1\\ 1&0&0&0&0&1&0&1\\ 0&1&0&0&1&0&1&0\\ 0&0&1&0&0&1&0&1\\ 0&0&0&1&1&0&1&0 \end{bmatrix}$$

3. Find the graphs whose incidence matrices are

(a) $$\begin{bmatrix}1&1&0&0&0\\0&1&1&0&0\\0&0&1&1&0\\0&0&0&1&1\\1&0&0&0&1\\0&1&0&0&1\\0&1&0&1&0\end{bmatrix};$$

(b) $$\begin{bmatrix}1&1&0&0&0&0&0&0\\1&0&1&0&0&0&0&0\\0&1&1&0&0&0&0&0\\0&1&0&1&0&0&0&0\\0&0&1&1&0&0&0&0\\0&1&0&0&0&1&0&0\\0&0&1&0&0&0&1&0\\0&0&0&0&1&1&0&0\\0&0&0&0&1&0&1&0\\0&0&0&0&0&1&1&0\\0&0&0&0&0&1&0&1\\0&0&0&0&0&0&1&1\end{bmatrix}$$

(c) $$\begin{bmatrix}1&1&0&0&0&0&0&0\\0&1&1&0&0&0&0&0\\0&0&1&1&0&0&0&0\\1&0&0&1&0&0&0&0\\0&0&0&0&1&1&0&0\\0&0&0&0&0&1&1&0\\0&0&0&0&0&0&1&1\\0&0&0&0&1&0&0&1\\1&0&0&0&1&0&0&0\\0&1&0&0&0&1&0&0\\0&0&1&0&0&0&1&1\\0&0&0&1&0&0&0&1\end{bmatrix}$$

****

Chapter 8

Graph Algorithms

****

An algorithm may be considered as a set of instructions for solving certain mathematical problems when followed step by step will lead to the solutions of that problem. Usually, an algorithm is first written in ordinary language and then converted in to flowcharts and finally, written in a detailed and precise language so that the machine can execute it.

An algorithm must not only do what it is supposed to do, but also must do it efficiently, in terms of the requirements of both memory and computation time requirements as the function of the size of the input. In our case, the input will be graphs with order n and size ϵ.

8.1 Spanning Tree Algorithm

A *spanning tree algorithm* is the one which yields a spanning tree from a given connected graph and spanning forest from a given disconnected graph. Let G be a graph (connected or disconnected) with n vertices and ϵ edges. Then, this algorithm has the following steps.

S-1 : If it has loops, then, the loops may be discarded from G and the vertices be labeled $1, 2, 3, \ldots, n$ and the graph be arranged in two linear arrays F and H (as in linear array listing explained earlier) such that the i-th edge has end vertices $f_i \in F$ and $h_i \in H$.

S-2 : Choose an arbitrary edge $e_1 = f_1h_1$ and set the component number $c = 1$.

S-3 : At the k-th stage, where $1 \leq k \leq \epsilon$, one of the following situations may arise:

(a) If for the edge e_k, neither f_k nor h_k is included in any of the trees constructed so far in G, that edge may be named as a new tree and its end vertices are given the component number c after incrementing the value of c by 1.

(b) If vertex f_k is in some tree $T_i; i = 1,2,3,\dots,c$ and h_k in tree $T_j; j = 1,2,3,\dots,c; i \neq j$, the k-th edge is used to join these two vertices and hence every vertex in T_j is now given the component number of T_i. The value of c is decremented by 1.

(c) If both vertices are in the same tree, the edge f_kh_k forms a fundamental cycle and is not considered further.

(d) If the vertex f_k is in a tree T_i and the vertex h_k is not in any tree, then add the edge f_kh_k is added to T_i by assigning the component number of T_i to h_k also. If the vertex f_k is not in any tree and the vertex h_k is in a tree T_j, then add the edge f_kh_k is added to T_j by assigning the component number of T_j to f_k also.

S-4 : Stop if some component number is assigned to all vertices of G.

For example, consider the graph given in Figure **??**. Consider the edge $e_1 = v_1v_2$ (where $v_1 \in F$ and $v_2 \in H$). Both vertices are not currently in any tree of G and hence by Step S-3(a) add ${}_1v_2$ to the tree T_1 and assign the component number 1 to v_1 and v_2.

Now, consider the vertex pairs $(v_1,v_3),(v_1,v_4),(v_1,v_5)$ and (v_1,v_6). For all these vertex pairs, exactly one vertex (v_1) is in the tree T_1 and as per Step-S-3(d), all these vertices can be added to the tree T_1 and assign the component number 1.

Next, for the vertex pair (v_2,v_8), only v_2 currently belongs to the tree T_1. So, by Step-S-3(d), we can add v_2v_8 to the tree T_1 and give the component number 1.

For the vertex pair (v_3,v_8), both v_3 and v_8 belong to the tree T_1 and hence by Step S-3(c), we can never consider v_3v_8 to any tree of G further.

Similarly, if we consider v_6v_{10} to the tree T_1 we cannot consider v_5v_{10} and if the edge v_8v_{12} is included in T_1, then the edge $v_{10}v_{12}$ can never be considered to any tree further.

For the vertex pairs (v_4,v_7) and (v_4,v_{11}), none of the vertices is in any tree so far and hence by Step S-3(a), they can be included to a new tree T_2 and assign the component number 2.

Now, if we consider the edge v_7v_{14} to the tree T_2, then, by Step-S3(c), the edge $v_{11}v_{14}$ cannot be taken to any tree further.

For the vertex pairs $(v_{14},v16)$ and (v_{14},v_{18}), only v_{14} belongs to a tree in G, and hence both edges can be added to T_2 with component number 2.

Similarly, for the vertex pairs (v_9,v_{13}) and (v_9,v_{15}), none of the vertices is in any tree so far and hence by Step S-3(a), they can be included to a new tree T_3 and assign the component number 3.

Now, if we consider the edge $v_{13}v_{17}$ to the tree T_3, then, by Step-S3(c), the edge $v_{15}v_{17}$ cannot be considered for inclusion in any tree further.

Then, the spanning forest (as G is disconnected) of the graph G is as given in Figure 8.1.

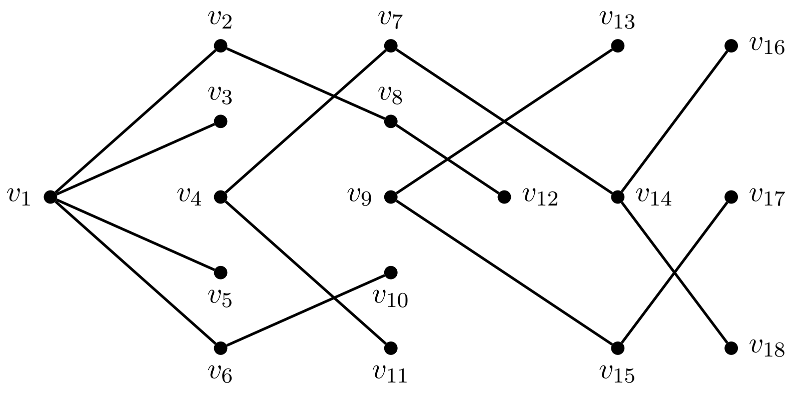


Figure 8.1: A spanning forest of a disconnected G

8.2 Minimal Spanning Tree Algorithms

Many optimisation problems amount to finding, in a suitable weighted graph, a certain type of subgraph with the minimum (or maximum) weight. In the following problems, we are looking for a spanning tree of G of the minimum weight. Such a tree is called a *minimal spanning tree* or an *optimal spanning tree* for G. Minimum spanning trees have many interesting applications such as the design of a network. There are several algorithms for computing the minimum spanning trees. They all, however, are based on the same underlying property about cuts in a graph, which we will refer to as the *light-edge property*. One of the most important algorithms among these is Kruskal's algorithm.

8.2.1 Kruskal's Algorithm

The procedure is explained in the following three steps.

(i) Choose an edge e_1, an edge of G, such that $w(e_1)$ is as small as possible and e_1 is not a loop.

(ii) If edges $e_1, e_2, \ldots, e_i$ have been chosen, then choose an edge e_{i+1}, not already, chosen, such that

  (a) the induced subgraph $G[\{e_1, \ldots, e_{i+1}\}]$ is acyclic; and

  (b) $w(e_{i+1})$ is as small as possible subject to Step-1.

(iii) If G has n vertices, stop after $n-1$ edges have been chosen. Otherwise, repeat Step 2.

A step by step illustration of Kruskal's Algorithm to find a minimal spanning tree of the graph given Figure 8.2 is given below.

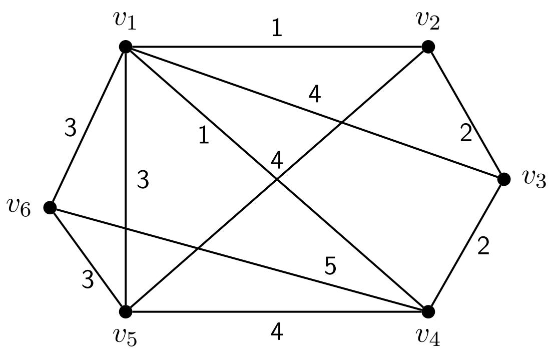


Figure 8.2: A weighted graph G

Figure 8.3 illustrates how to determine a minimal spanning tree of the weighted graph G given above using Kruskal's algorithm.

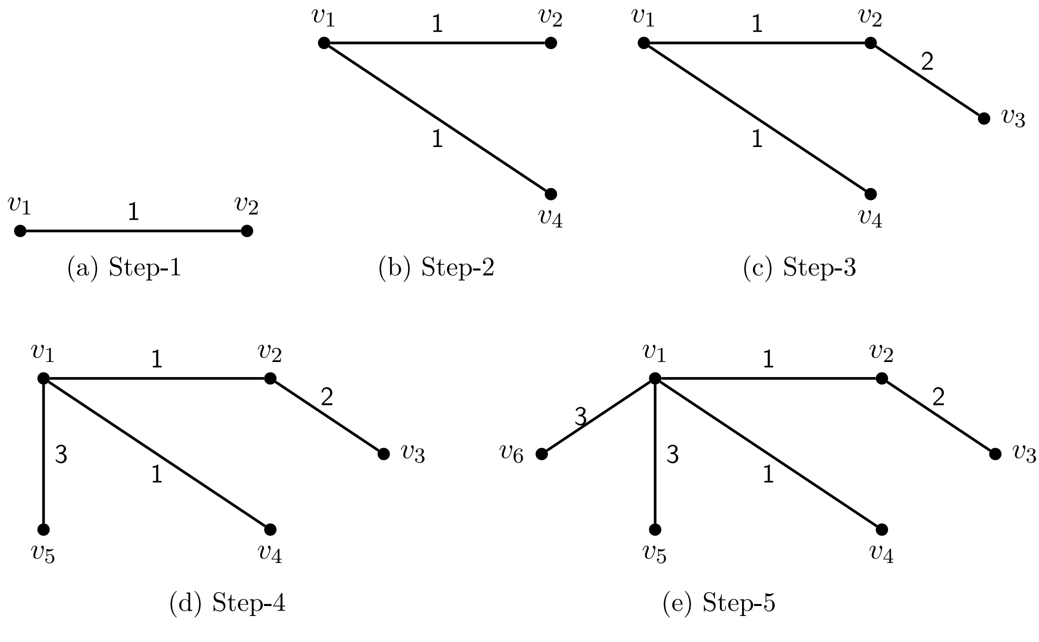


Figure 8.3: Illustration to Kruskal's Algorithm

Theorem 8.2.1. *Kruskal's algorithm yields the shortest spanning tree of the graph G under consideration.*

Proof. In this algorithm, we first choose an edge of G which has smallest weight among the edges of G which are not loops. Next, again avoiding loops, we choose from the remaining edges one of smallest weight possible which does not form a cycle with the edge already chosen. We repeat this process taking edges of smallest weight among those not already chosen, provided no cycle is formed with those that have been chosen. This condition ensures that we get acyclic subgraphs of G with weights as the minimum as possible. If the graph has n vertices we stop after having chosen $n-1$ edges. This is because by Theorem 5.2.8, any acyclic graph with n vertices and $n-1$ edges is connected and hence is a tree. Therefore, these $n-1$ edges form a minimal spanning tree of G. □

8.2.2 Prim's Algorithm

This algorithm involves the following steps.

(i) Choose any vertex v_1 of G.

(ii) Choose an edge $e_1 = v_1v_2$ of G such that $v_2 \neq v_1$ and e_1 has the smallest weight among the edges of G incident with v_1.

(iii) If edges $e_1, e_2, \ldots, e_i$ have been chosen involving end points $v_1, v_2, \ldots, v_{i+1}$, choose an edge $e_{i+1} = v_jv_k$ with $v_j \in \{v_1, \ldots, v_{i+1}\}$ and $v_k \notin \{v_1, \ldots, v_{i+1}\}$ such that e_{i+1} has the smallest weight among the edges of G with precisely one end in $\{v_1, \ldots, v_{i+1}\}$.

(iv) Stop after $n-1$ edges have been chosen. Otherwise, repeat Step-3.

A step by step illustration of Prim's Algorithm to find a minimal spanning tree of the graph given Figure 8.2 is given below.

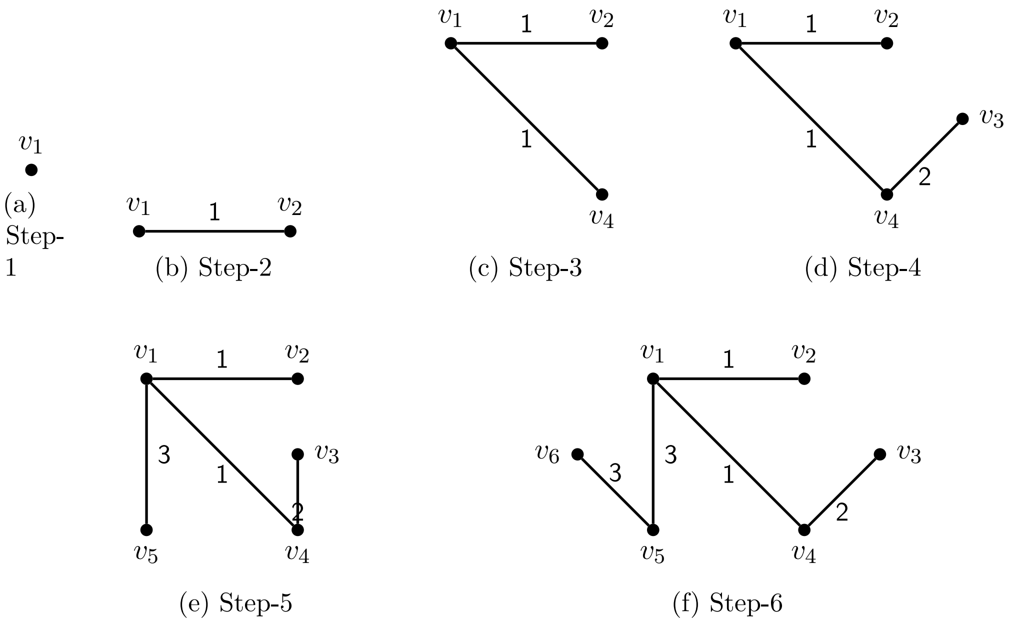


Figure 8.4: Illustration to Prim's Algorithm

Theorem 8.2.2. *Prim's algorithm yields the shortest spanning tree of the graph G under consideration.*

Proof. In this algorithm for finding a minimal spanning tree, we first choose a (any) vertex v_1 of the connected graph G. We next choose one of the edges incident with v_1 having the smallest weight in G which is not a loop, say $e_1 = v_1v_2$. Next, we choose an edge of smallest weight in G which is incident with either v_1 or v_2, but with the other end vertex neither of these. That is, we choose $e_2 = v_iv_3$, where $i = 1$ or $i = 2$. We repeat this process of taking edges of smallest weight one of whose ends is a vertex previously chosen and the other end becoming involved for the first time, until we have chosen $n-1$ edges (assuming the graph has n vertices). Clearly, this condition ensures that at each stage the subgraph develops satisfying the connectedness acyclic conditions and has the weight as minimum as possible. At the final stage, we have involved each of the n vertices of G and by the construction the resulting subgraph is connected, acyclic and has the weight as minimum as possible. Hence, the subgraph of G thus obtained is a minimal spanning tree. □

1. Both algorithms have the same output. But, Kruskal's algorithm begins with forest and merge into a tree, while Prim's algorithm always stays as a tree.
2. Both algorithms have the same complexity $O(N \log N)$, as both involve sorting edges for lowest weight.
3. Kruskal's Algorithm usually has to check for the weights of the whole edges of the graph at each iteration and hence works best, if the number of edges is kept to a minimum. But, Prim's algorithm doesn't need to see the weights of all edges of the graph at once. It can deal with it one piece at a time. It also doesn't need to worry if adding an edge will create a cycle since this algorithm deals primarily with the nodes, and not the edges.
4. The running time of Kruskal's algorithm is $O(\epsilon \log n)$, while the running time of Prim's algorithm is $O(\epsilon + \log n)$.

8.3 Shortest Path Problems

Let G be a graph and let s, t be two specified vertices of G. In shortest spanning tree problems, we have to a path from s to t, if there is any, which uses the least number of edges. Such a path, if it exists, is called a *shortest path* from s to t.

8.3.1 Dijkstra's Algorithm

The *Dijkstra's algorithm* is an algorithm for finding the shortest paths between nodes in a graph, which may represent (for example, road networks). It was conceived by computer scientist *Edsger W. Dijkstra* in 1956. Dijkstra's original variant found the shortest path between two nodes, but a more common variant fixes a single node as the "source" vertex and finds shortest paths from the source to all other vertices in the graph, producing a shortest-path tree. Given a path P from vertex s to vertex t in a weighted graph G, we define the length of P to be the sum of the weights of its edges.

The steps involved in Dijkstra's Algorithm are as follows:

1. Set $\lambda(s) = 0$ and for all vertices $v \neq s$, $\lambda(v) = \infty$. Set $T = V$, the vertex set of G. (We will think of T as the set of vertices *uncoloured.*)
2. Let u be a vertex in T for which $\lambda(u)$ is the minimum.
3. If $u = t$, stop.
4. For every edge $e = uv$ incident with u, if $v \in T$ and $\lambda(v) > \lambda(u) + w(uv)$, change the value of $\lambda(v)$ to $\lambda(u) + w(uv)$ (That is, given an edge $e = uv$ from an *uncoloured* vertex v to u, change $\lambda(v)$ to $\min\{\lambda(v), \lambda(u) + w(uv)\}$.
5. Change T to $T - \{u\}$ and go to Step-2, (That is, *colour* u and then go back to Step-2 to find an *uncoloured* vertex with the minimum label).

Consider the weighted graph in Figure 8.5.

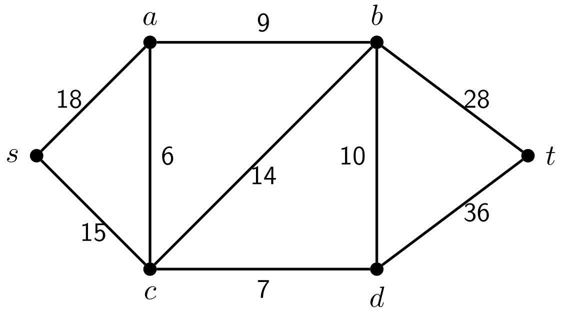


Figure 8.5: A weighted graph G

A step by step illustration of Dijkstra's Algorithm for the graph given in Figure ?? is explained as follows:

Step 1. The initial labeling is given by:

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	∞	∞	∞	∞	∞
T	{s,	a,	b,	c,	d,	$t\}$

Step 2. $u = s$ has $\lambda(u)$ a minimum (with value 0).

Step 4. There are two edges incident with u, namely sa and sc. Both a and c are in T, i.e., they are not yet coloured. $\lambda(a) = \infty > 18 = 0 + 18 = \lambda(s) + w(sa)$. Hence, $\lambda(a)$ becomes 18. Similarly, $\lambda(c)$ becomes 15.

Step 5. T becomes $T - \{s\}$, i.e., we colour s. Thus we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	∞	15	∞	∞
T	{	a,	b,	c,	d,	$t\}$

Step 2. $u = c$ has $\lambda(u)$ a minimum and u in T (with $\lambda(u) = 15$).

Step 4. There are 3 edges cv incident with $u = c$ having v in T, namely ca, cb, cd. Considering the edge ca, we have $\lambda(a) = 18 < 21 = 15 + 6 = \lambda(c) + w(ca)$ and hence $\lambda(a)$ remains as 18. Considering the edge cb, we have $\lambda(b) = \infty > 29 = 15 + 14 =$

$\lambda(c)+w(cb)$. So, $\lambda(b)$ becomes 29. Similarly, considering the edge cd, we have $\lambda(d)$ becomes $15+7=22$.

Step 5. T becomes $T-\{c\}$, i.e., we colour c. Thus, we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	29	15	22	∞
T	$\{$	a,	b,		d,	$t\}$

Step 2. $u=a$ has $\lambda(u)$ a minimum and for u in T (with $\lambda(u)=18$).

Step 4. There is only one edge av incident with $u=a$ having v in T, namely ab. Then, $\lambda(b)=29>27=18+9=\lambda(a)+w(ab)$. So, $\lambda(b)$ becomes 27.

Step 5. T becomes $T-\{a\}$, i.e., we colour a. Thus, we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	27	15	22	∞
T	$\{$		b,		d,	$t\}$

Step 2. $u=d$ has $\lambda(u)$ minimum for u in T (with $\lambda(u)=22$).

Step 4. There are two edges dv incident with $u=d$ having v in T, namely db and dt. Considering the edge db, we have $\lambda(b)=27<32=22+10=\lambda(d)+w(db)$ and hence $\lambda(b)$ remains as 27. For the edge dt, we have $\lambda(t)=\infty>58=22+36=\lambda(d)+w(dt)$. Hence, $\lambda(t)$ becomes 58.

Step 5. T becomes $T-\{d\}$, i.e., we colour d. Thus, we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	27	15	22	58
T	$\{$		b,			$t\}$

Step 2. $u=b$ has $\lambda(u)$ minimum for u in T (with $\lambda(u)=27$).

Step 4. There is only one edge bv for v in T, namely bt. $\lambda(t)=58>55=27+28=\lambda(b)+w(bt)$ so $\lambda(t)$ becomes 55.

Step 5. T becomes $T-ab$, i.e., we colour b. Thus, we have

vertex v	s	a	b	c	d	t
$\lambda(v)$	0	18	27	15	22	55
T	$\{$					$t\}$

Step 2. $u=t$, the only choice. Step 3. Stop.

All steps, we did above can be written in a single table as follows:

Note that Dijkstra's algorithm does not work with negative edges. The problem is that when we consider the closest vertex v not in the visited set, its shortest path may be through only the visited set and then extended by one edge out of the visited set to v.

Problem 8.3.1. Using Dijkstra's Algorithm, find the shortest path between v_1 and v_6 in the following graph:

Steps	s	a	b	c	d	t	T					
1	0	∞	∞	∞	∞	∞	s	a	b	c	d	t
2	0	18	∞	15	∞	∞		a	b	c	d	t
3	0	18	29	15	22	∞		a	b		d	t
4	0	18	27	15	22	∞			b		d	t
5	0	18	27	15	22	58			b			t
6	0	18	27	15	22	55						t

Table 8.1: Combined table of Dijkstra's Procedure.

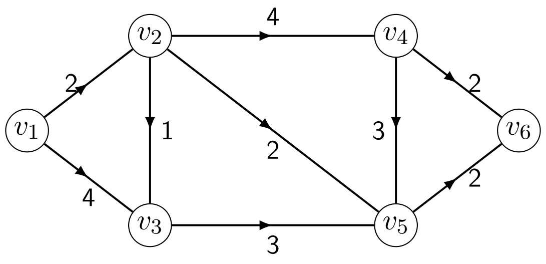


Figure 8.6: A weighted graph G

vertex v	v_1	v_2	v_3	v_4	v_5	v_6
$\lambda(v)$	0	∞	∞	∞	∞	∞
T	v_1	v_2	v_3	v_4	v_5	v_6

Solution: Solution: The initial labeling is given by:

Let $u = v_1$, as $\lambda(v_1)$ is the minimum (with value 0) in the above table. Now, v_2 and v_3 are adjacent to u. Therefore,

$$\begin{aligned} \lambda(v_2) &= \min\{\lambda(v_2), \lambda(v_1) + w(v_1v_2)\} \\ &= \min\{\infty, 0+2\} = 2 \\ \lambda(v_3) &= \min\{\lambda(v_3), \lambda(v_1) + w(v_1v_3)\} \\ &= \min\{\infty, 0+4\} = 4 \end{aligned}$$

Therefore, we have

vertex v	v_1	v_2	v_3	v_4	v_5	v_6
$\lambda(v)$	0	2	4	∞	∞	∞
T		v_2	v_3	v_4	v_5	v_6

Let $u = v_2$, as $\lambda(v_2)$ is the minimum (with value 2) in the above table. Now, v_3, v_4 and v_5 are adjacent to u. Therefore, we have

$$\begin{aligned}\lambda(v_3) &= \min\{\lambda(v_3), \lambda(v_2)+w(v_2v_3)\}\\ &= \min\{4, 2+1\} = 3\\ \lambda(v_4) &= \min\{\lambda(v_4), \lambda(v_2)+w(v_2v_4)\}\\ &= \min\{\infty, 2+4\} = 6\\ \lambda(v_5) &= \min\{\lambda(v_5), \lambda(v_2)+w(v_2v_5)\}\\ &= \min\{\infty, 2+2\} = 4\end{aligned}$$

Hence,

vertex v	v_1	v_2	v_3	v_4	v_5	v_6
$\lambda(v)$	0	2	3	6	4	∞
T			v_3	v_4	v_5	v_6

Then, let Let $u = v_3$, as $\lambda(v_3)$ is the minimum (with value 3) in the above table. The vertex v_5 is adjacent to u. Therefore, we have

$$\begin{aligned}\lambda(v_5) &= \min\{\lambda(v_5), \lambda(v_3)+w(v_3v_5)\}\\ &= \min\{4, 3+3\} = 4\end{aligned}$$

Hence, the above table will not change. Next, let $u = v_5$, as $\lambda(v_5)$ is the minimum (with value 4) in the above table. Now, v_6 is adjacent to u. Therefore, we have

$$\begin{aligned}\lambda(v_6) &= \min\{\lambda(v_6), \lambda(v_4)+w(v_4v_6)\}\\ &= \min\{6, 2+6\} = 6\end{aligned}$$

Therefore, we have

vertex v	v_1	v_2	v_3	v_4	v_5	v_6
$\lambda(v)$	0	2	3	6	4	6
T				v_4		v_6

Next, let $u = v_4$, as $\lambda(v_4)$ is the minimum (with value 6) in the above table. Now, v_6 is adjacent to u. Therefore, we have

$$\lambda(v_6) = \min\{\lambda(v_6), \lambda(v_5)+w(v_5v_6)\}$$

$$= \min\{6, 6+2\} = 6$$

vertex v	v_1	v_2	v_3	v_4	v_5	v_6
$\lambda(v)$	0	2	3	6	4	6
T						v_6

Next, let $u = v_6$, as v_6 is the only element in T in the above table. Therefore, the procedure terminates.

Hence, the combined table of the procedure will be obtained as follows:

Steps	v_1	v_2	v_3	v_4	v_5	v_6	T					
1	0	∞	∞	∞	∞	∞	v_1	v_2	v_3	v_4	v_5	v_6
2	0	2	4	∞	∞	∞		v_2	v_3	v_4	v_5	v_6
3	0	2	3	6	4	∞			v_3	v_4	v_5	v_6
4	0	2	3	6	4	6				v_4	v_5	v_6
5	0	2	3	6	4	6				v_4		v_6
6	0	2	3	6	4	6						v_6

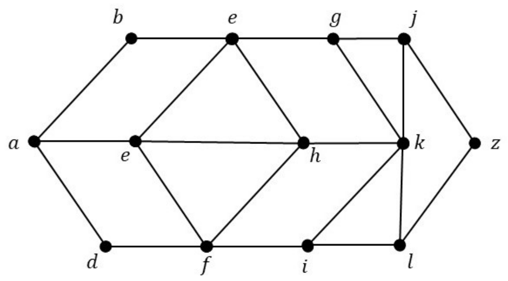
Table 8.2: Combined table of Dijkstra's Procedure.

Therefore, the shortest path from v_1 to v_6 is $v_1 - v_2 - v_5 - v_6$ and the minimal weight of this $v_1 - v_6$ path is 6. □

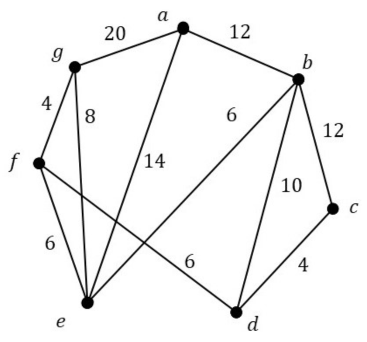
Theorem 8.3.2. *For any graph G, Dijkstra's algorithm computes the distance (weight) of shortest paths from a fixed vertex s to all other vertices in G.*

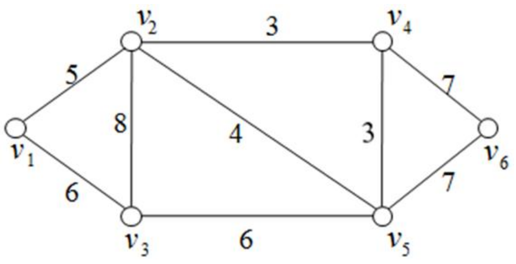
8.4 Exercise Problems

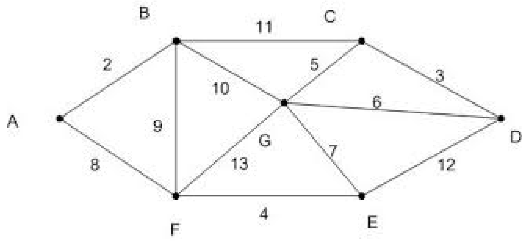
1. Describe different ways of representing a graph in computer along with illustrations.
2. Carry out the BFS algorithm for the following graph to find the length of a shortest path from vertex a to vertex z, an example of such a shortest path, and the number of shortest paths from a to z.

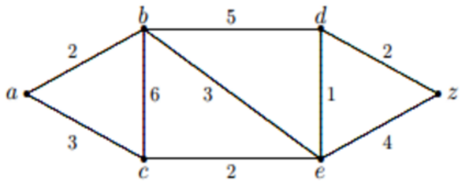


3. Use Dijkstra's algorithm on the connected weighted graphs of the following figures to find the length of the shortest paths from the vertex a to each of the other vertices and to give examples of such paths.

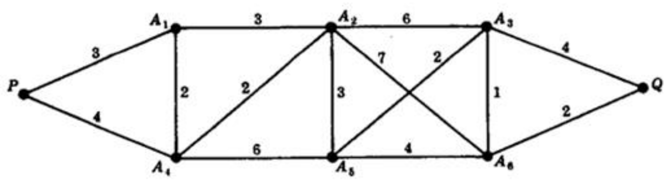


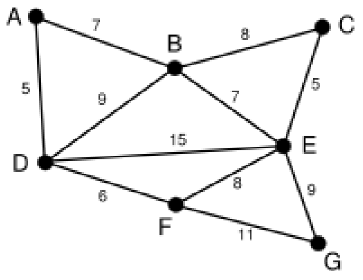






4. Use Kruskal's algorithm and Prim's algorithm to find minimal spanning trees of the following graphs:
5. Write an algorithm to construct a graph such that a given matrix as its incidence graph.





6. Write an algorithm to construct a graph such that a given matrix as its adjacency graph.

****

Bibliography

****

[1] G. Agnarsson and R. Greenlaw, (2007). *Graph theory: Modeling, applications & algorithms*, Pearson Education, New Delhi.

[2] J. Aldous and R.J. Wilson, (2004). *Graphs and applications*, Springer-Verlag, London.

[3] S. Arumugam and S. Ramachandran, (2015), *Invitation to graph theory*, Scitech Publ., Kolkata, India.

[4] R. Balakrishnan and K. Ranganathan, (2012). *A textbook of graph theory*, Springer, New York.

[5] V.K. Balakrishnan, (1997). *Graph theory*, McGrawhill, New York.

[6] A. Benjamin, G. Chartrant and P. Zhang, (2015). *The fascinating world of graph theory*, Princeton Uty. Press, New Jersey.

[7] S. Bornholdt, H.G. Schuster, (2003). *Handbook of graphs and networks*, Wiley-VCH, Weinheim.

[8] C. Berge, (1979). *Graphs and hypergraphs*, North Holland, Amsterdam.

[9] C. Berge, (2001). *The theory of graphs*, Dover Pub.

[10] B. Bollabás, (1998). *Modern graph theory*, Springer Int. Edn., New York.

[11] J.A. Bondy and U.S.R Murty, (2008). *Graph theory*, Springer.

[12] J.A. Bondy and U.S.R Murty, (1976). *Graph theory with applications*, North-Holland, New York.

[13] A. Brandstädt, V. B. Le and J. P. Spinrad, (1999). *Graph classes: A survey*, SIAM, Philadelphia.

[14] F.Buckley and F.Harary, (1990). *Distances in graphs*, Addison-Wesley Publishing Company.

[15] G. Chartrand and P. Zhang, (2005). *Introduction to graph theory*, McGraw-Hill Inc.

[16] G. Chartrand and L. Lesniak, (1996). *Graphs and digraphs*, CRC Press.

[17] W.K. Chen, (1997). *Graph theory and its engineering applications*, World Scientific Publications, Singapore.

[18] J. Clark and D.A. Holton, (1995). *A first look at graph theory*, Allied Publishers Ltd., Mumbai, India.

[19] N. Deo, (2016). *Graph Theory with application to engineering and computer science*, Prentice Hall of India Pvt. Ltd., India.

[20] R. Diestel, (2010). *Graph theory*, Springer-Verlag, New York.

[21] S. Even, (2012). *Graph algorithms*, Cambridge University Press, New York.

[22] J.C. Fournier, (2009). *Graph theory with applications*, John Wiley & Sons, Inc., New York.

[23] H. Fleishner, (1990). *Eulerian graphs and related topics*, North Holland-Amsterdam.

[24] M.C. Golumbic and I.B.A. Hartman, (2005). *Graph theory, combinatorics and algorithms*, Springer.

[25] R.P. Grimaldi, (2014). *Discrete and combinatorial mathematics*, Pearson Education., New Delhi.

[26] J. Gross and J. Yellen, (1999). *Graph theory and its applications*, CRC Press.

[27] J. Gross and J. Yellen, (2004). *Handbook of graph theory*, CRC Press.

[28] F. Harary, (2001). *Graph theory*, Narosa Publ. House, New Delhi.

[29] F. Harary and E. M. Palmer, (1973). *Graphical enumeration*, Academic Press Inc.,

[30] J. B. Jensen and G. Gutin, (2007). *Digraphs: Theory, algorithms and applications*, Springer- Verlag.

[31] K. D. Joshi, (2003). *Applied discrete structures*, New Age International, New Delhi.

[32] D Joyner, M.V. Nguyen, N. Cohen, (2012). *Algorithmic graph theory*, `http://code.google.com/p/graphbook/`.

[33] D Marcus, (2008). *Graph theory a problem oriented approach*, MAA Textbooks, USA.

[34] T.A. McKee and F.R. McMorris, (1999). *Topics in intersection graph theory*, SIAM Monographs on Discrete Mathematics and Application, SIAM, Philadelphia.

[35] O. Ore, *Theory of Graphs*, (1962). American Math. Soc. Colloquium Pub., XXXV III.

[36] M.S. Rahman, (2017). *Basic graph theoy*, Springer, Switzerland.

[37] F.S. Roberts, (1978). *Graph theory and its practical applications to problems of society*, SIAM, Philadelphia.

[38] K. Rosen, (2011). *Discrete mathematics and its applications*, Tata McGraw Hill.

[39] R.J. Trudeau, (1993), *Introduction to graph theory*, Dover Publ., New York.

[40] S.S. Sarkar (2003). *A text book of discrete mathematics*, S. Chand & Co., New Delhi.

[41] G. Sethuraman, R. Balakrishnan, and R.J. Wilson, (2004). *Graph theory and its applications*, Narosa Pub. House, New Delhi.

[42] G.S. Singh, (2013). *Graph theory*, Prentice Hall of India, New Delhi.

[43] N.K. Sudev, (2017). *Lecture notes on graph theory*; Centre for Studies in Discrete Mathematics, Thrissur, India.

[44] N.K. Sudev, (2018). *Topics in graph theory*; Owl Books, Thiruvananthapuram, India.

[45] R.R. Stoll, (1979). *Set theory and Logic*, Dover pub., New York.

[46] K. Thulasiraman, M.N.S. Swamy, (). *Graphs: Theory and algorithms*, John Wiley & Sons, New York.

[47] W.D. Wallis, (2007). *A beginner's guide to graph theory*, Birkhäuser Boston.

[48] D.B. West, (2001). *Introduction to graph theory*, Pearson Education Inc., Delhi.

****

List of Figures

****

List of Tables

****

Index

****