Module - 5

Chapter 9

Directed Graphs and Graphs

The last two chapters of this book are concerned with *Graph Theory.* In this chapter we first introduce and illustrate the conepts of *directed graphs* and *graphs*. Then we present some basic material oncerned with graphs and related concepts*.

9.1 Directed Graphs

Look at the diagram shown below. This diagram consists of four vertices A, B, C, D and three edges AB, CD, CA with *directions attached to them*, the directions being indicated by arrows. Because of attaching directions to the edges, the edge AB has to be interpreted as an edge *from the vertex A to the vertex B* and it cannot be written as BA. Similarly, the edge CD is from C to D and cannot be written as DC, and the edge CA is from C to A and cannot be written as AC. Thus, here, the edges AB, CD, CA are *directed edges***.

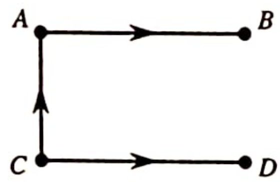


Figure 9.1

The directed edge AB is determined by the vertices A and B in that order and may therefore be represented by the *ordered pair* (A, B). Similarly, the directed edges CD and CA may be represented by the ordered pairs (C, D) and (C, A) respectively. Thus, the diagram in Figure 9.1 consists of a nonempty set of vertices, namely $\{A, B, C, D\}$, and a set of directed edges represented by *ordered pairs* of vertices taken from this set, namely $\{(A, B), (C, D), (C, A)\}$. Such a diagram is called a diagram of a *directed graph* (or a *diagraph* for brevity).

The formal definition of a directed graph is given below.

---

*In graph theory, the definitions, notation and terminology are not yet standardized; they generally vary from one author to the other. The reader has to keep this in mind while using different books.

**If one wishes, these edges may also be denoted by $\overrightarrow{AB}$, $\overrightarrow{CD}$, $\overrightarrow{CA}$ to emphasize that these are actually directed edges.

Definition of a Directed graph

A directed graph (or a digraph) is a pair (V, E), where V is a nonempty set and E is a set of ordered pairs of elements taken from the set V.

For a directed graph (V, E), the elements of V are called ***vertices*** (*points* or *nodes*) and the elements of E are called ***directed edges***. The set V is called the ***vertex set*** and the set E is called the ***directed edge set***.

For brevity in terminology, a directed edge is often referred to as just an "edge". Similarly, the directed edge set is referred to as the "edge set."

The directed graph (V, E) is also denoted $D = (V, E)$, or $D = D(V, E)$, or just D when there is no ambiguity.

A geometrical figure that depicts a directed graph is called a *diagram of the directed graph*.

Thus, Figure 9.1 is a diagram of the directed graph for which the vertex set is

$$V = \{A, B, C, D\}$$

and the edge set is

$$E = \{AB, CD, CA\} = \{(A, B), (C, D), (C, A)\}.$$

Figure 9.2 depicts the directed graph for which the vertex set is $V = \{A, B, C, D\}$ and the edge set is $E = \{AB, CD, AC\} = \{(A, B), (C, D), (A, C)\}$.

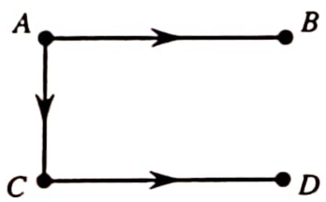


Figure 9.2

It has to be noted that the directed graph depicted in Figure 9.2 is *not* the same as the directed graph depicted in Figure 9.1. Although both of these two directed graphs have the same vertex set, their directed edge sets are different; where as the directed graph in Figure 9.1 has CA as a directed edge, the directed graph in Figure 9.2 does not have CA as a directed edge – it has AC as a directed edge, and the directed edges AC and CA are not one and the same.

It has to be mentioned that in a diagram of a directed graph the directed edges need not be straight line segments; they can be curved lines (arcs) also. For example, a direced edge AB of a directed graph can be represented by an arbitrary arc drawn from the vertex A to the vertex B as shown in Figure 9.3.

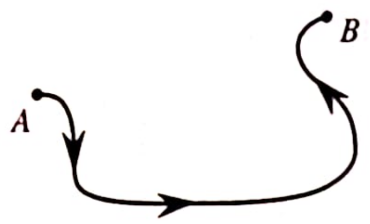


Figure 9.3

Thus, a directed graph can be depicted in a diagram in more than one way. For brevity in terminology, a diagram of a directed graph itself is often referred to as a directed graph.

Vertices of directed graphs are denoted by upper or lower case letters, like A, B, $C, \ldots$, u, $v, \ldots$, or, by letters with suffixes appended to them, like $v_1, v_2, v_3, \ldots$.

As illustrated in Figure 9.1, every directed edge of a digraph (directed graph) is determined by two vertices of the digraph – a vertex from which it begins and a vertex at which it ends. Thus, if AB is a directed edge of a digraph D, then it is understood that this directed edge begins at the vertex A of D and terminates at the vertex B of D. Here, we say that A is the ***initial vertex*** and B is the ***terminal vertex*** of AB. Equivalently, we say that AB is incident *out of A* and incident *into B*.

It should be mentioned that for a directed edge (in a digraph) the initial vertex and the terminal vertex need not be different. A directed edge beginning and ending at the same vertex A is denoted by AA or (A, A) and is called a ***directed loop***. The directed edge shown in Figure 9.4 is a directed loop which begins and ends at the vertex A.

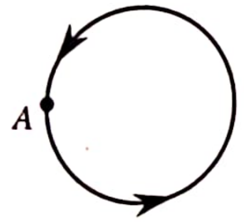


Figure 9.4

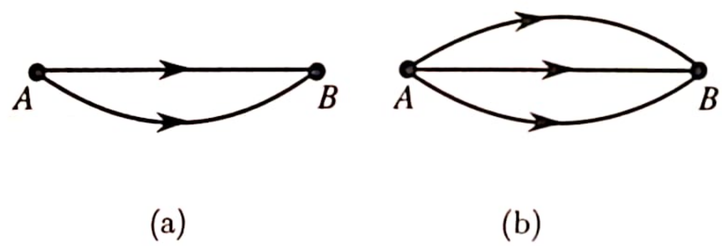


Figure 9.5

A digraph can have more than one directed edge having the same initial vertex and the same terminal vertex. Two directed edges having the same initial vertex and the same terminal vertex are called ***parallel directed edges***. Two parallel directed edges are shown in Figure 9.5(a). Two or more directed edges having the same initial vertex and the same terminal vertex are called ***multiple directed edges****. Three multiple edges are shown in Figure 9.5(b).

A vertex of a digraph which is neither an initial vertex nor a terminal vertex of any directed edge is called an ***isolated vertex*** of the digraph. A non-isolated vertex happens to be an initial vertex or a terminal vertex for a (some) directed edge. A non-isolated vertex which is not a

*When a digraph contains multiple edges, the edge set becomes a ***multiple set***; by a multiple set we mean a set in which repetition of elements is also taken into account.

terminal vertex for any directed edge is called a ***source*** and a non-isolated vertex which is not an initial vertex for any directed edge is called a ***sink***.

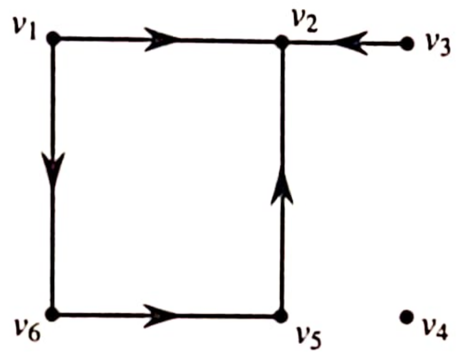


Figure 9.6

In the digraph shown in Figure 9.6, the vertex v_4 is an isolated vertex, the vertices v_1 and v_3 are sources and the vertex v_2 is a sink. The vertices v_5 and v_6 do not belong to any of these categories.

In-degree and Out-degree

If v is a vertex of a digraph D, the number of edges for which v is the initial vertex is called the ***out-going degree*** or the ***out-degree*** of v and the number of edges for which v is the terminal vertex is called the ***incoming degree*** or the ***in-degree*** of v. The out-degree of v is denoted by $d^+(v)$ or $od(v)$ and the in-degree of v is denoted by $d^-(v)$ or $id(v)$.

If follows that (i) $d^+(v) = 0$ if v is a sink, (ii) $d^-(v) = 0$ if v is a source, and (iii) $d^+(v) = d^-(v) = 0$ if v is an isolated vertex.

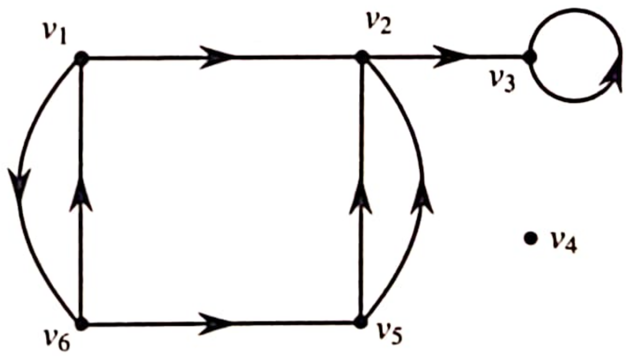


Figure 9.7

For the digraph shown in Figure 9.7, the out-degrees and the in-degrees of the vertices are as given below:

$$d^+(v_1) = 2, \qquad d^-(v_1) = 1,$$
$$d^+(v_2) = 1, \qquad d^-(v_2) = 3,$$

$$d^+(v_3) = 1, \qquad d^-(v_3) = 2,$$
$$d^+(v_4) = 0, \qquad d^-(v_4) = 0,$$
$$d^+(v_5) = 2, \qquad d^-(v_5) = 1,$$
$$d^+(v_6) = 2, \qquad d^-(v_6) = 1.$$

We note that, in the above digraph, there is a directed loop at the vertex v_3 and this loop contributes a count 1 to each of $d^+(v_3)$ and $d^-(v_3)$.

We further observe that the above digraph has 6 vertices and 8 edges and that the sums of the out-degrees and in-degrees of its vertices are

$$\sum_{i=1}^{6} d^+(v_i) = 8, \quad \sum_{i=1}^{6} d^-(v_i) = 8.$$

This illustrates the following property common to all digraphs. This property is referred to as the *First Theorem of the Digraph Theory*.

Property : *In every digraph D, the sum of the out-degrees of all vertices is equal to the sum of the in-degrees of all vertices, each sum being equal to the number of edges in D.*

Proof: Suppose D has n vertices $v_1, v_2, \ldots, v_n$ and m edges. Let r_1 be the number of edges going out of v_1, r_2 be the number of edges going out of v_2, and so on. Then

$$d^+(v_1) = r_1, \quad d^+(v_2) = r_2, \ldots, \quad d^+(v_n) = r_n.$$

Since every edge terminates at some vertex and since there are m edges, we should have

$$r_1 + r_2 + \cdots + r_n = m.$$

Accordingly,

$$d^+(v_1) + d^+(v_2) + \cdots + d^+(v_n) = r_1 + r_2 + \cdots + r_n = m.$$

Similarly, if s_1 is the number of edges coming into v_1, s_2 is the number of edges coming into v_2, and so on, we get

$$d^-(v_1) + d^-(v_2) + \cdots + d^-(v_n) = s_1 + s_2 + \cdots + s_n = m.$$

Thus,

$$\sum_{i=1}^{n} d^+(v_i) = \sum_{i=1}^{n} d^-(v_i) = m.$$

This completes the proof. •

Example 1 *Find the in-degrees and the out-degrees of the vertices of the digraph shown in Figure* 9.8.

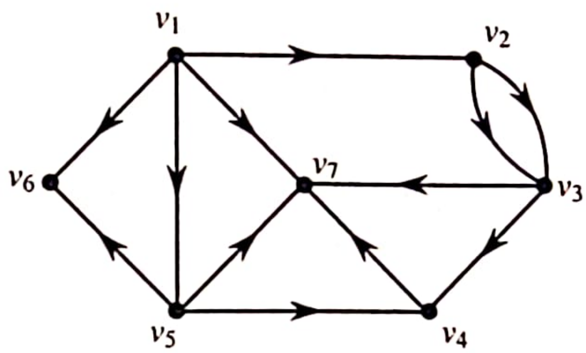


Figure 9.8

► The given digraph has 7 vertices and 12 directed edges. The out-degree of a vertex is got by counting the number of edges that go out of the vertex and the in-degree of a vertex is got by counting the number of edges that end at the vertex. Thus, we obtain the following data:

Vertex:	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Out-degree:	4	2	2	1	3	0	0
In-degree:	0	1	2	2	1	2	4

This table gives the out-degrees and in-degrees of all vertices. We note that v_1 is a source and v_6 and v_7 are sinks.

We also check that

sum of out-degrees = sum of in-degrees = 12 = No. of edges. ■

Example 2 *Let D be a digraph with an odd number of vertices. If each vertex of D has an odd out-degree, prove that D has an odd number of vertices with odd in-degrees.*

► Let $v_1, v_2, \ldots, v_n$ be the n vertices of D, where n is odd. Also, let m be the number of edges in D. Then, we have

$$d^+(v_1) + d^+(v_2) + \cdots + d^+(v_n) = m \tag{i}$$

and

$$d^-(v_1) + d^-(v_2) + \cdots + d^-(v_n) = m \tag{ii}$$

If each vertex has odd out-degree, then the left hand side of (i) is a sum of n odd numbers. Since n is odd, this sum must also be odd. Thus, m is odd.

Let k be the number of vertices with odd in-degrees. Then $n - k$ number of vertices have even in-degrees. Without loss of generality, let us take $v_1, v_2, \ldots v_k$ to be the vertices with odd in-degrees and $v_{k+1}, v_{k+2}, \ldots v_n$ to be the vertices with even in-degrees. Then, expression (ii) may be rewritten as

$$\sum_{i=1}^{k} d^-(v_i) + \sum_{i=k+1}^{n} d^-(v_i) = m \tag{iii}$$

Now, the second sum on the left hand side of this expression is even. Also, m is odd. Therefore, the first sum must be odd. That is

$$d^-(v_1) + d^-(v_2) + \cdots + d^-(v_k) = \text{odd} \tag{iv}$$

But, each of $d^-(v_1)$, $d^-(v_2), \ldots d^-(v_k)$ is odd. Therefore, the number of terms in the left hand side of (iv) must be odd. That is, k is odd. This proves the required result. ∎

Exercises

1. Write down the vertex set and the directed edge set of each of the following digraphs.

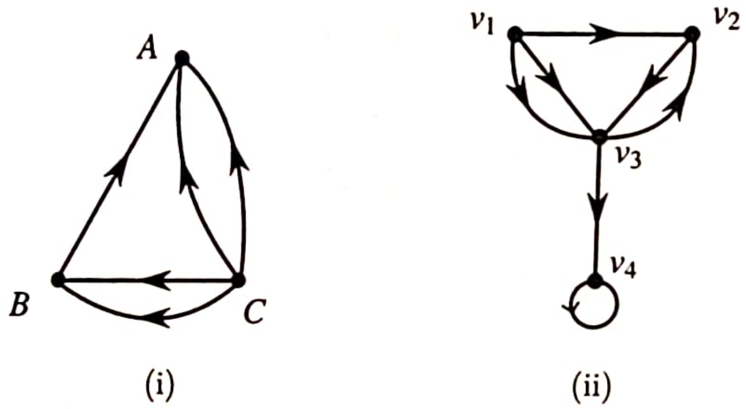


Figure 9.9

2. For the digraph shown in Figure 9.6, determine the out-degrees and in-degrees of all the vertices.
3. For the digraphs of Exercise 1 above, determine the out-degrees and in-degrees of all the vertices.
4. Let D be the digraph whose vertex set is $V = \{v_1, v_2, v_3, v_4, v_5\}$ and the directed edge set is

$$E = \{(v_1, v_4), (v_2, v_3), (v_3, v_5), (v_4, v_2), (v_4, v_4), (v_4, v_5), (v_5, v_1)\}.$$

Write down a diagram of D and indicate the out-degrees and in-degrees of all the vertices.

5. Verify the First theorem of Digraph theory for (i) the digraphs shown in Figures 9.6 and 9.9, and (ii) the digraphs shown below:

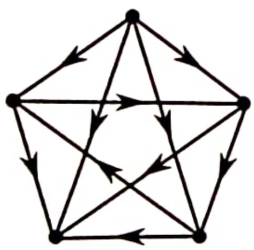


Figure 9.10

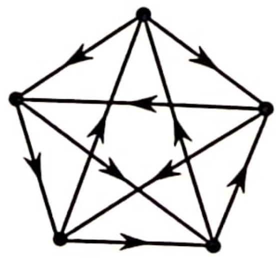


Figure 9.11

Answers

1. (i) Vertex set is $V = \{A, B, C\}$ and the directed edge set is

$E = \{(B, A), (C, A), (C, A), (C, B), (C, B)\}$.

(ii) Vertex set is $V = \{v_1, v_2, v_3, v_4\}$ and the directed edge set is

$E = \{(v_1, v_2), (v_1, v_3), (v_1, v_3), (v_2, v_3), (v_3, v_2), (v_3, v_4), (v_4, v_4)\}$.

2. $d^-(v_1) = 0, d^-(v_2) = 3, d^-(v_3) = 0, d^-(v_4) = 0, d^-(v_5) = 1, d^-(v_6) = 1.$

$d^+(v_1) = 2, d^+(v_2) = 0, d^+(v_3) = 1, d^+(v_4) = 0, d^+(v_5) = 1, d^+(v_6) = 1.$

3. (i) $d^+(A) = 0, d^+(B) = 1, d^+(C) = 4, \quad d^-(A) = 3, d^-(B) = 2, d^-(C) = 0.$

(ii) $d^+(v_1) = 3, d^+(v_2) = 1, d^+(v_3) = 2, d^+(v_4) = 1, \quad d^-(v_1) = 0, d^-(v_2) = 2, d^-(v_3) = 3, d^-(v_4) = 2.$

4.

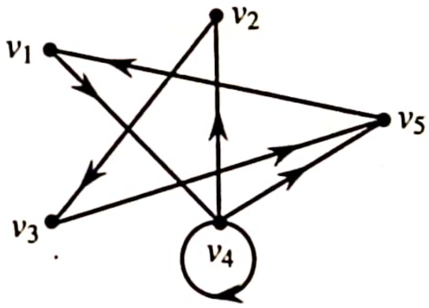


Figure 9.12

Vertices	v_1	v_2	v_3	v_4	v_5
d^+	1	1	1	3	1
d^-	1	1	1	2	2

9.2 Graphs

Consider again the diagram in Figure 9.1. Let us redraw this diagram by *dropping* the arrows present in the edges. The rewritten diagram appears as shown below.

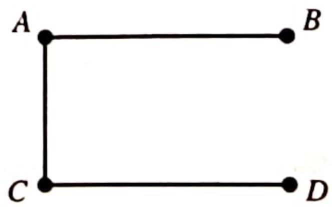


Figure 9.13

We say that this (rewritten) diagram represents the diagram of an *underlying graph* of the directed graph shown in Figure 9.1. This underlying graph has the same vertices and the same edges as the original directed graph; but its edges are all *undirected.*

We observe that Figure 9.13 depicts a diagram of the underlying graph of the directed graph shown in Figure 9.2 as well.

The underlying graph of a directed graph is an example of what is called an *undirected graph.* Like a directed graph, an undirected graph also consists of a nonempty set of vertices and a set of edges, but the edges are all "undirected". An edge in a undirected graph is determined and represented by an *unordered pair* of vertices. For example, in the undirected graph shown in Figure 9.13, the edge AB is determined by the vertices A and B and is represented by the unordered pair $\{A, B\} = \{B, A\}$.* For brevity in terminology, an undirected graph is referred to as just *a graph* (– the adjective "undirected" being understood).†

The formal definition of a graph is given below.

Definition of a Graph

A graph is a pair (V, E), where V is a nonempty set and E is a set of unordered pairs of elements taken from the set V.

For a graph (V, E), the elements of V are called ***vertices*** (or *points* or *nodes*) and the elements of E are called ***undirected edges*** or just ***edges***. The set V is called the ***vertex set*** and the set E is called the ***edge set***.

The graph (V, E) is also denoted by $G = (V, E)$ or $G = G(V, E)$ or just G when there is no ambiguity.

A geometrical figure that depicts a graph is called a *diagram of the graph*. Thus, Figure 9.13 is a diagram of the graph for which the vertex set is $V = \{A, B, C, D\}$ and the edge set is $E = \{AB, AC, CD\} = \{\{A, B\}, \{A, C\}, \{C, D\}\}$.

According to the definition of a graph/digraph, the vertex set in a graph/digraph has to be non-empty. Thus, *a graph/digraph must contain at least one vertex*. But, the edge set can be empty. This means that a graph/digraph need not contain any edge.

A graph/digraph containing no edges is called a ***null graph***. A null graph with only one vertex is called a ***trivial graph***. Figure 9.14 depicts a null graph with three vertices.

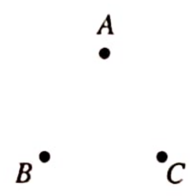


Figure 9.14

*We use the notation (A, B) to denote the *ordered pair* of A and B in that order and the notation $\{A, B\} = \{B, A\}$ for the *unordered pair* of A and B.

†The terms *directed graph* and *digraph* mean one and the same. The terms *undirected graph* and *graph* mean one and the same.

In diagrams depicting non-null graphs, an edge is represented by a line. The line may be *straight* or *curved, long* or *short*. The important thing is that the diagram should represent the vertices and edges correctly, irrespective of whether the edges are drawn as straight lines or as curves.

The way one draws a diagram of a graph is basically immaterial. There can be more than one diagram for the same graph. For instance, the two diagrams in Figure 9.15(a) and 9.15(b) *look* different, yet they represent the same graph since each coveys the same information, namely that the graph has four vertices A, B, C, D with AB, AC, AD, BC and CD as edges. For brevity in terminology, a diagram of a graph itself is often referred to as a graph.

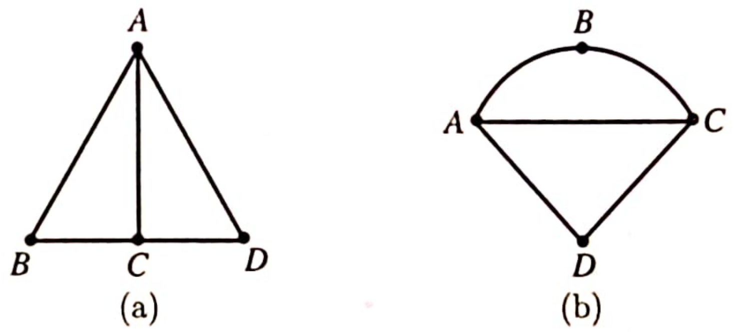


Figure 9.15

The definition of a graph/digraph does not impose any upper limits for the number of vertices and the number of edges. Thus, a graph/digraph can have infinitely many vertices and/or infinitely many edges. A graph/digraph with only a finite number of vertices as well as only a finite number of edges is called a ***finite graph/digraph***; otherwise, it is called an *infinite graph/digraph*. In what follows, we will be concerned only with finite graphs/digraphs. Accordingly, *by a graph/digraph we will mean only a finite graph/digraph.*

In the following discussions (in this Chapter and in Chapter 10), our major interest will be towards graphs. Only when it is needed, a digraph will be brought in. All the definitions and results that hold for graphs can be extended to digraphs, but with appropriate modifications in terminology.

Order and Size

The number of vertices in a (finite) graph is called the ***order*** of the graph and the number of edges in it is called its ***size***. In other words, for a graph $G = (V, E)$, the cardinality of the set V, namely $|V|$, is called the order of G and the cardinality of the set E, namely $|E|$, is called the size of G. A graph of order n and size m is called a ***(n, m) graph***. Thus, the graph depicted in Figure 9.15(a) is a (4.5) graph. A null graph with n vertices is a $(n, 0)$ graph.

End vertices, loop, multiple edges

Generally, the vertices of a graph are denoted by A, B, C, etc., or v_1, v_2, v_3, etc. When it is convenient, we denote the edges of a graph by e_1, e_2, e_3, and so on. If v_i and v_j denote two vertices of a graph and if e_k denotes an edge joining v_i and v_j, then v_i and v_j are called the ***end***

vertices (or *end points*) of e_k. This is symbolically written as $e_k = \{v_i, v_j\} = v_iv_j$. For example, in the graph shown in Figure 9.15(a), suppose we denote the edges AB, BC, AC, AD and DC by e_1, e_2, e_3, e_4 and e_5 respectively. Then e_1 joins A and B; that is, $e_1 = \{A, B\} = AB$ so that A and B are the end vertices of e_1. Similarly, e_2 joins B and C; that is, $e_2 = \{B, C\} = BC$, so that B and C are the end vertices of e_2, and so on.

Now, consider the graph shown in Figure 9.16. We note that this graph consists of four vertices v_1, v_2, v_3, v_4, and six edges e_1, e_2, e_3, e_4, e_5, e_6. Although the edges e_2 and e_3 seem to intersect (cross over) in the figure,* their point of intersection (even when it exists) is *not* a vertex of the graph. We observe that the edges e_1, e_2, e_3 have distinct end vertices, but the edge e_4 has the same vertex v_3 as both of its end vertices; that is, $e_4 = \{v_3, v_3\}$. An edge such as e_4 is called a ***loop***. We also observe that both of the edges e_5 and e_6 have the same end vertices v_1, v_4; that is, $e_5 = \{v_1, v_4\}$ and $e_6 = \{v_1, v_4\}$. Edges such as these are called ***parallel edges***. If in a graph there are two or more edges with the same end vertices, the edges are called ***multiple edges.***

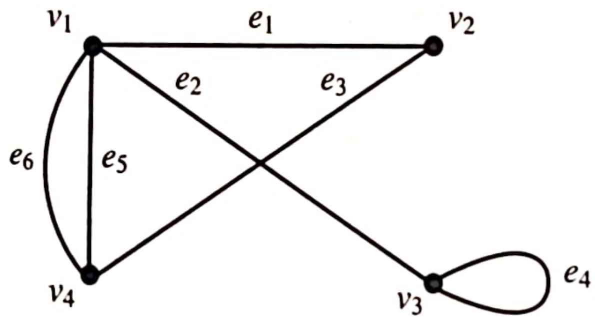


Figure 9.16

Simple graph, Multigraph, General graph

A graph which does not contain loops and multiple edges is called a ***simple graph***. A graph which does not contain a loop is called a ***loop-free*** graph. A graph which contains multiple edges but no loops is called a ***multigraph***†. A graph which contains multiple edges or loops (or both) is called a ***general graph***.

Figure 9.13 represents a simple graph and Figure 9.16 represents a general graph. Figure 9.17 represents a multigraph.

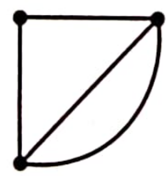


Figure 9.17

*If the vertices are in different planes, e_2 and e_3 may not intersect.

†In a multigraph, the edge set is a multiset.

In our discussions here, *by a graph we will mean a general graph*, unless otherwise mentioned.

In may situations, the names assigned to vertices are inconsequential. In such situations, we do not assign any names to the vertices. Such graphs are called ***unlabeled graphs***. On the other hand, if names are assigned to vertices of a graph (for some specific purpose or upon certain grounds), the graph is called a ***labeled graph***. For example, the graphs shown in Figures 9.13 to 9.16 are labeled graphs. The graph shown in Figure 9.17 is unlabeled.

Incidence

When a vertex v of a graph G is an end vertex of an edge e of the graph G, we say that the edge e is ***incident on*** (or *to*) the vertex v. Since every edge has two end vertices, every edge is incident on two vertices, one at each end. The two end vertices are coincident if the edge is a loop.

When an edge e is incident on a vertex v, we also say that v is *incident with* e. Note that whereas an edge is incident only on two vertices (namely its end vertices), a vertex may be incident with any number of edges.

Two *non-parallel* edges are said to be ***adjacent edges*** if they are incident on a common vertex (that is, if they have a vertex in common). Two vertices are said to be ***adjacent vertices*** (or ***neighbours***) if there is an edge joining them.

In the graph shown in Figure 9.18, A and B are adjacent vertices and e_1 and e_2 are adjacent edges. But, A and C are not adjacent vertices, and e_1 and e_3 are not adjacent edges.

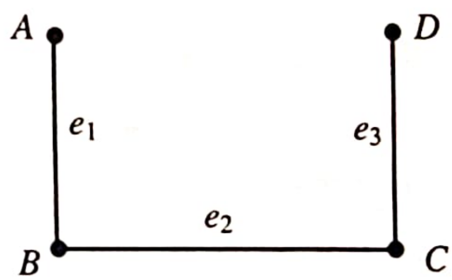


Figure 9.18

Complete graph

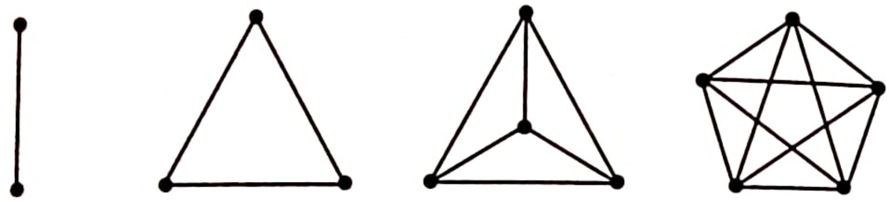
A *simple graph* of order ≥ 2 in which there is an edge between *every pair* of vertices is called a ***complete graph*** (or a *full graph*).

In other words, *a complete graph is a simple graph of order ≥ 2 in which every pair of distinct vertices are adjacent.*

A complete graph with $n\ (\geq 2)$ vertices is denoted by K_n.

Complete graphs with two, three, four and five vertices are shown in Figures 9.19(a) to 9.19(d) respectively. Of these complete graphs, the complete graph with five vertices, namely

K_5 (shown in Figure 9.19(d)), is of great importance. This graph is called the ***Kuratowski's first graph***.*



(a) K_2 (b) K_3 (c) K_4 (d) K_5

Figure 9.19

Bipartite graph

Suppose a *simple graph* G is such that its vertex set V is the union of two of its mutually *disjoint* nonempty subsets V_1 and V_2 which are such that *each* edge in G joins a vertex in V_1 and a vertex in V_2. Then G is called a ***bipartite graph***. If E is the edge set of this graph, the graph is denoted by $G = (V_1, V_2; E)$, or $G = G(V_1, V_2; E)$. The sets V_1 and V_2 are called ***bipartites*** (or *partitions*) of the vertex set V.

For example, consider the graph G shown in Figure 9.20 for which the vertex set is $V = \{A, B, C, P, Q, R, S\}$ and the edge set is $E = \{AP, AQ, AR, BR, CQ, CS\}$. Note that the set V is the union of two of its subsets $V_1 = \{A, B, C\}$ and $V_2 = \{P, Q, R, S\}$ which are such that (i) V_1 and V_2 are disjoint, (ii) every edge in G joins a vertex in V_1 and a vertex in V_2, (iii) G contains no edge that joins two vertices both of which are in V_1 or V_2. This graph is a bipartic graph with $V_1 = \{A, B, C\}$ and $V_2 = \{P, Q, R, S\}$ as bipartites.

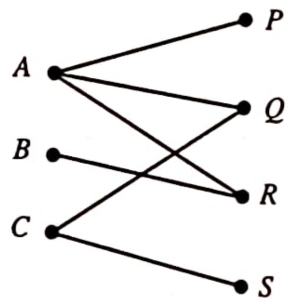


Figure 9.20

Complete Bipartite graph

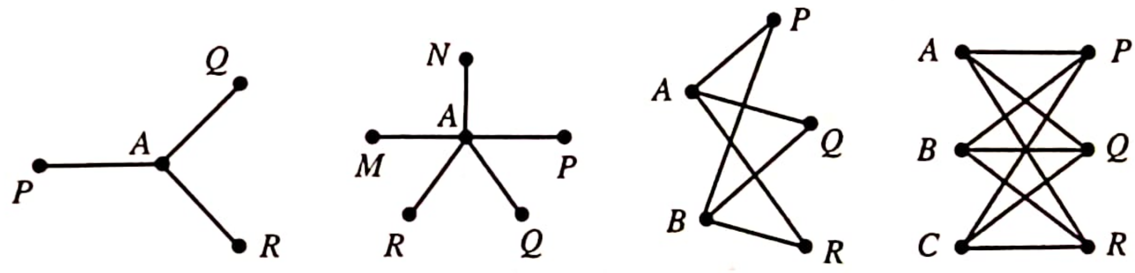
A bipartite graph $G = (V_1, V_2; E)$ is called a ***complete bipartite graph*** if there is an edge between every vertex in V_1 and every vertex in V_2.

The bipartite graph shown in Figure 9.20 is *not* a complete bipartite graph. Observe, for example, that the graph does not contain an edge joining A and S.

*Named after the Polish mathematician Kasimir Kuratowski.

A complete bipartite graph $G = (V_1, V_2; E)$ in which the bipartites V_1 and V_2 contain r and s vertices respectively, with $r \leq s$, is denoted by $K_{r,s}$. In this graph, each of r vertices in V_1 is joined to each of s vertices in V_2. Thus, $K_{r,s}$ has $r + s$ vertices and rs edges; that is $K_{r,s}$ is of order $r + s$ and of size rs; it is therefore a $(r + s, rs)$ graph.

Figures 9.21(a) to 9.21(d) depict some complete bipartite graphs. Observe that in Figure 9.21(a), the bipartites are $V_1 = \{A\}$ and $V_2 = \{P, Q, R\}$; the vertex A is joined to each of the vertices P, Q, R by an edge. In Figure 9.21(b), the bipartites are $V_1 = \{A\}$ and $V_2 = \{M, N, P, Q, R\}$; the vertex A is joined to each of the vertices M, N, P, Q, R by an edge.



(a) : $K_{1,3}$ (b) : $K_{1,5}$ (c) : $K_{2,3}$ (d) : $K_{3,3}$

Figure 9.21

In Figure 9.21(c), the bipartites are $V_1 = \{A, B\}$ and $V_2 = \{P, Q, R\}$; each of the vertices A and B is joined to each of the vertices P, Q, R by an edge. In Figure 9.21(d), the bipartites are $V_1 = \{A, B, C\}$ and $V_2 = \{P, Q, R\}$; each of the vertices A, B, C is joined to each of the vertices P, Q, R.

Of these complete bipartite graphs, the graph $K_{3,3}$ shown in Figure 9.21(d) is of great importance. This is known as the ***Kuratowski's second graph***.

It is to be noted that a bipartite graph G is not a complete graph even if G is a complete bipartite graph. Because, in such a graph, there exists no edge between two vertices if they belong to the same bipartite.

Example 1 *Draw a diagram of the graph $G = (V, E)$ in each of the following cases:*

(i) $V = \{A, B, C, D\}$, $E = \{AB, AC, AD, CD\}$

(ii) $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{v_1v_2, v_1v_3, v_2v_3, v_4v_5\}$

(iii) $V = \{P, Q, R, S, T\}$, $E = \{PS, QR, QS\}$

(iv) $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $E = \{v_1v_4, v_1v_6, v_4v_6, v_3v_2, v_3v_5, v_2v_5\}$

▶ The required diagrams are shown below:

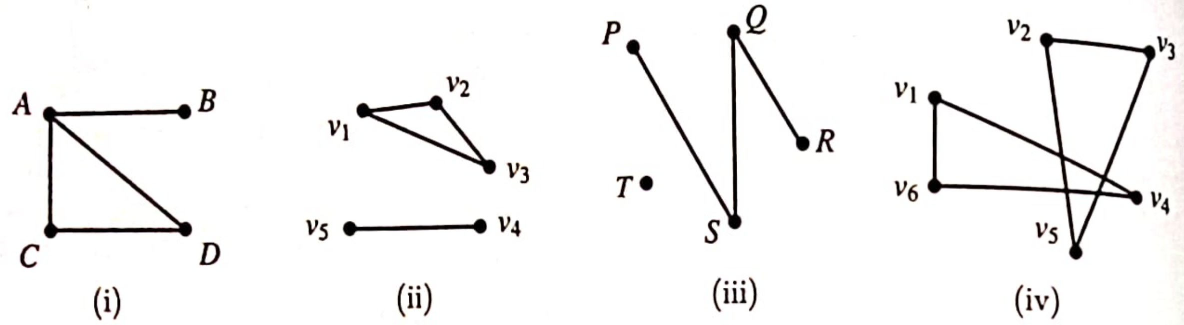


Figure 9.22

∎

Example 2 *Let P, Q, R, S, T represent five cricket teams. Suppose that the teams P, Q, R have played one game with each other, and the teams P, S, T have played one game with each other. Represent this situation in a graph.*

Hence determine (i) *the teams that have not played with each other, and* (ii) *the number of games played by each team.*

▶ Let the teams be represented by vertices and an edge represent the playing. Then the graph representing the given situation is as shown in Figure 9.23:

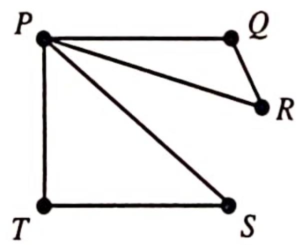


Figure 9.23

We observe that there is no edge between Q and S, between Q and T, between R and S, and between R and T. Therefore, the teams Q and S, Q and T, R and S, and R and T have not played with each other.

From the graph, we note that two edges are incident on each of the vertices Q, R, S, T and four edges are incident on P. Thus, the teams Q, R, S, T have played two games each and the team P has played four games. ∎

Example 3 *Which of the following is a complete graph?*

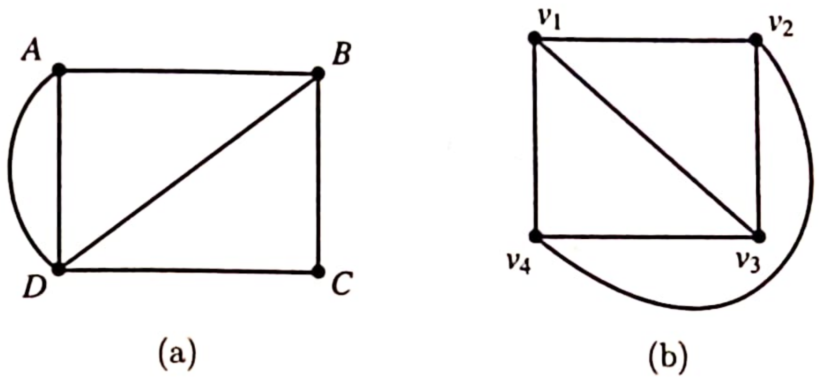


Figure 9.24

► The first of the graphs is *not* complete. It is not simple on the one hand and there is no edge between A and C on the other hand. The second of the graphs is complete. It is a simple graph and there is an edge between every pair of vertices. ■

Example 4 *If $G = G(V, E)$ is a simple graph, prove that* $2|E| \leq |V|^2 - |V|$.

► Each edge of a graph is determined by a pair of vertices. In a simple graph there occur no multiple edges. As such, in a simple graph, the number of edges *cannot exceed* the number of pairs of vertices. The number of pairs of vertices that can be choosen from n vertices is (from the theory of combinations)

$$^nC_2 = \frac{n!}{(n-2)!2!} = \frac{1}{2}n(n-1).$$

Thus, for a simple graph with $n\,(\geq 2)$ vertices, the number of edges cannot exceed $\frac{1}{2}n(n-1)$. Accordingly, if a simple graph $G = G(V, E)$ has n vertices and m edges, then $m \leq \frac{1}{2}n(n-1)$, or, $2m \leq n^2 - n$; that is : $2|E| = |V|^2 - |V|$. ■

Remark: If a graph is not simple, it can have any number of edges (because, in such a graph multiple edges and loops are allowed).

Example 5 *Show that a complete graph with n vertices, namely K_n, has $\frac{1}{2}n(n-1)$ edges.*

► In a complete graph, there exists exactly one edge between every pair of vertices. As such, the number of edges in a complete graph is equal to the number of pairs of vertices. If the number of vertices is n, then the number of pairs of vertices is

$$^nC_2 = \frac{n!}{(n-2)!2!} = \frac{1}{2}n(n-1)$$

Thus, the number of edges in a complete graph with n vertices is $\frac{1}{2}n(n-1)$. ■

Note: For another proof of this result, see Example 13, Section 9.2.1.

Example 6 *Show that a simple graph of order $n = 4$ and size $m = 7$ and a complete graph of order $n = 4$ and size $m = 5$ do not exist.*

► For $n = 4$, we have

$$\frac{1}{2}n(n-1) = \frac{1}{2} \times 4 \times 3 = 6.$$

Since $m = 7$ exceeds this number, a simple graph of order $n = 4$ and size $m = 7$ does not exist.

Similarly, since $m = 5$ is not equal to $\frac{1}{2}n(n-1) = 6$, a complete graph of order 4 and size $m = 5$ does not exist. ∎

Example 7 *Which of the following is a bipartite graph?*

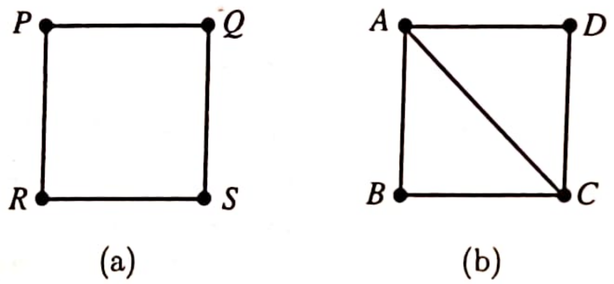


Figure 9.25

► The first of the graphs is a bipartite graph, with $V_1 = \{P, S\}$ and $V_2 = \{Q, R\}$ as the bipartites. The second graph is *not* a bipartite graph. (Why ?) ∎

Example 8 (a) *How many vertices and how many edges are there in the complete bipartite graphs $K_{4,7}$ and $K_{7,11}$?*

(b) *If the graph $K_{r,12}$ has 72 edges, what is r?*

► Recall that the complete bipartite graph $K_{r,s}$ has $r + s$ vertices and rs edges. Accordingly:

(a) The graph $K_{4,7}$ has $4 + 7 = 11$ vertices and $4 \times 7 = 28$ edges, and the graph $K_{7,11}$ has 18 vertices and 77 edges.

(b) If the graph $K_{r,12}$ has 72 edges, we have $12r = 72$ so that $r = 6$. ∎

Example 9 *Let $G = (V, E)$ be a simple graph of order $|V| = n$ and size $|E| = m$. If G is a bipartite graph, prove that* $4m \leq n^2$.

► Let V_1 and V_2 be the bipartites of G, with $|V_1| = r$ and $|V_2| = s$. Since $|V| = n$, we should have $r + s = n$, so that $r = n - s$ and $s = n - r$.

The graph G has the maximum number of edges when each of the r vertices in V_1 is joined by an edge to each of the s vertices in V_2, and this maximum is equal to rs. This means that $|E| = m \leq rs$.

When n is even, we may express rs as

$$rs = r(n-r) = rn - r^2 = \left(\frac{n}{2}\right)^2 - \left(r - \frac{n}{2}\right)^2$$

Evidently, rs is maximum when $r = (n/2)$, the maximum being $(n/2)^2$. Thus, when n is even, we have

$$m \leq rs \leq \left(\frac{n}{2}\right)^2.$$

When n is odd, we may express rs as

$$rs = r(n-r) = \frac{(n-1)(n+1)}{4} - \left(r - \frac{n-1}{2}\right)\left(r - \frac{n+1}{2}\right)$$

Evidently, rs is maximum when $r = (n-1)/2$ or $r = (n+1)/2$, the maximum being $(n-1)(n+1)/4$. Thus, when n is odd, we have

$$m \leq rs \leq \frac{(n-1)(n+1)}{4} = \frac{n^2-1}{4} \leq \frac{n^2}{4} = \left(\frac{n}{2}\right)^2$$

Accordingly, in both of the possible cases (n is even or odd), we have

$$m \leq \left(\frac{n}{2}\right)^2, \quad \text{or} \quad 4m \leq n^2.$$ ■

Example 10 *Show that a simple graph of order $n = 4$ and size $m = 5$ cannot be a bipartite graph.*

► Here, $4m = 20$ and $n^2 = 16$, so that $4m > n^2$. Therefore, the given simple graph cannot be a bipartite graph. ■

Exercises

1. Indicate the order and size of each of the graphs shown below.

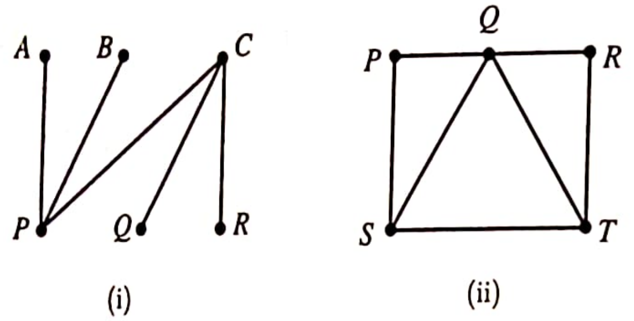


Figure 9.26

2. Identify the adjacent vertices in the graphs of the preceding exercise.

3. Identify the adjacent vertices and adjacent edges in the graph shown in Figure 9.16.

4. Which of the following graphs is a simple graph? a multigraph? a general graph?

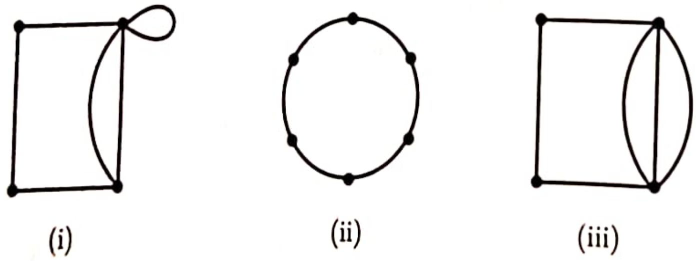


Figure 9.27

5. Which of the following are complete graphs?

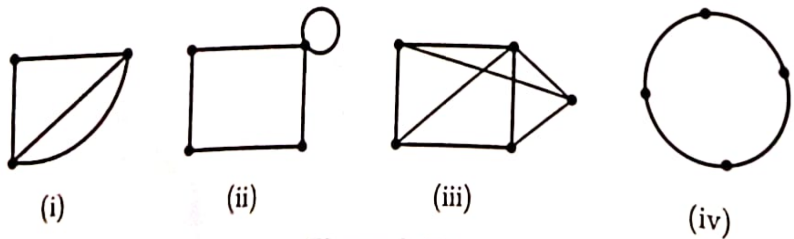


Figure 9.28

Scanned by CamScanner

6. Identify the graph shown below:

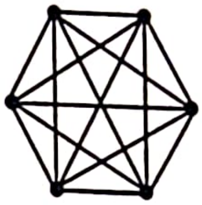


Figure 9.29

7. Verify that the following are bipartite graphs. What are their bipartites?

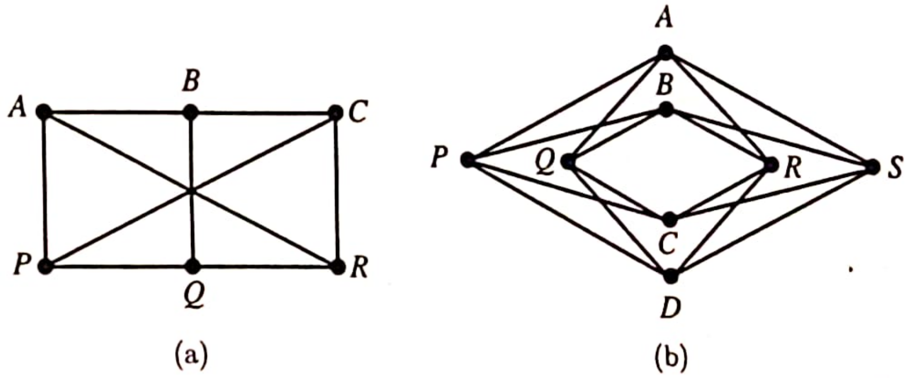


Figure 9.30

8. Which of the graphs shown below are bipartite graphs?

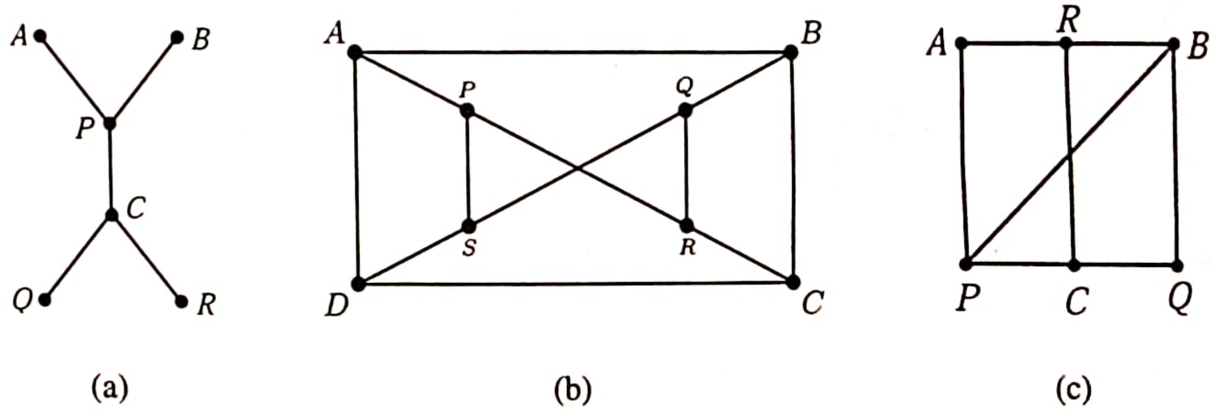


Figure 9.31

9. Company X has offices in cities B, D and K; company Y in cities B and M; company Z in cities C and M. Represent this situation by a bipartite graph. Is this a complete bipartite graph?

10. State whether the following graphs can exist or cannot exist.

    (1) Simple graph of order 3 and size 2.

    (2) Simple graph of order 5 and size 12.

    (3) Complete graph of order 5 and size 10.

    (4) Bipartite graph of order 4 and size 3.

    (5) Bipartite graph of order 3 and size 4.

    (6) Complete bipartite graph of order 4 and size 4.

Answers

1. (i) There are 6 vertices and 5 edges; therefore, the order is 6 and size is 5.

   (ii) There are 5 vertices and 7 edges; therefore, the order is 5 and size is 7.

2. (i) $\{A, P\}$, $\{P, B\}$, $\{P, C\}$, $\{C, Q\}$, $\{C, R\}$ are *pairs* of adjacent vertices.

   (ii) $\{P, Q\}$, $\{P, S\}$, $\{Q, S\}$, $\{Q, R\}$, $\{Q, T\}$, $\{R, T\}$, $\{S, T\}$ are *pairs* of adjacent vertices.

3. Adjacent vertices: v_1 and v_2, v_1 and v_3, v_1 and v_4, v_2 and v_4.

   Adjacent edges: e_1 and e_2, e_1 and e_3, e_1 and e_5, e_1 and e_6,

   e_2 and e_4, e_2 and e_5, e_2 and e_6, e_3 and e_5,

   e_3 and e_6.

4. (i) general graph (ii) simple graph, (iii) multigraph.

5. None is a complete graph.

6. The complete graph K_6.

7. (a) $V_1 = \{A, C, Q\}$, $V_2 = \{B, P, R\}$ (b) $V_1 = \{A, B, C, D\}$, $V_2 = \{P, Q, R, S\}$

8. (a) and (c) are bipartite graphs with $\{A, B, C\}$ and $\{P, Q, R\}$ as bipartites. (b) is not a bipartite graph.

9.

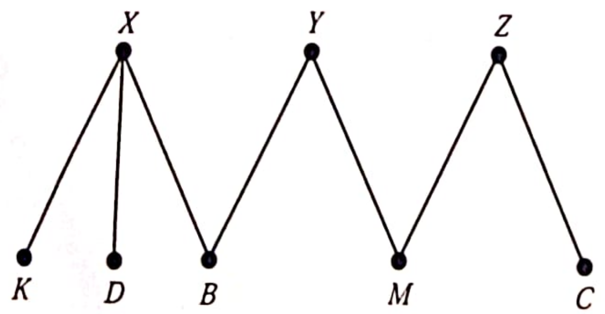


Figure 9.32

This is a not a complete bipartite graph.

10. (1), (3), (4), (6) : can exist

(2), (5) : cannot exist

9.2.1 Vertex degree and Handshaking property

Let $G = (V, E)$ be a graph and v be a vertex of G. Then, the number of edges of G that are incident on v (that is, the number of edges that join v to other vertices of G) *with the loops counted twice* is called the ***degree*** of the vertex v and is denoted by deg(v), or $d(v)$.*

The degrees of all vertices of a graph arranged in non-decreasing order is called the ***degree sequence*** of the graph. Also, the *minimum* of the degrees of vertices of a graph is called the ***degree of the graph***.

For example, the degrees of vertices of the graph shown in Figure 9.33 are as given below.

$$d(v_1) = 3, \quad d(v_2) = 4, \quad d(v_3) = 4, \quad d(v_4) = 3.$$

Therefore, the degree sequence of this graph is 3, 3, 4, 4, and the degree of the graph is 3.

Note that the loop at v_3 is counted twice for determining the degree of v_3.

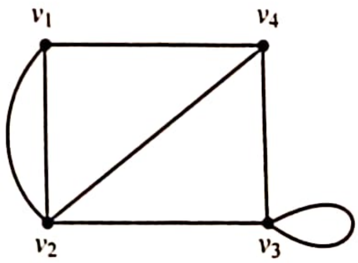


Figure 9.33

Isolated Vertex, Pendant Vertex

A vertex in a graph which is not an end vertex of any edge of the graph is called an ***isolated vertex***. Obviously, a vertex is an isolated vertex if and only if its degree is zero.

A vertex of degree 1 is called a ***pendant vertex***. An edge incident on a pendant vertex is called a ***pendant edge***.

In the graph shown in Figure 9.34, the vertices v_4 and v_6 are isolated vertices, v_5 and v_7 are pendant vertices and the edges e_4 and e_5 are pendant edges.

---

*Since, in a graph, all edges are undirected, no distinction is made between the in-degree and the out-degree. If G is an underlying graph of a directed graph D, then G and D have the same vertex set V and for any $v \in V$, we have $d(v) = id(v) + od(v)$.

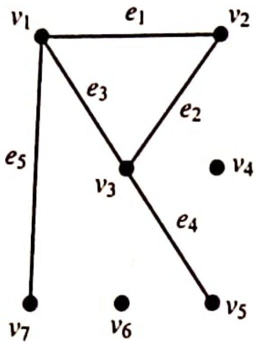


Figure 9.34

As mentioned before, a null graph contains no edges. It therefore follows that *in a null graph every vertex is an isolated vertex.*

Regular graph

A graph in which all the vertices are of the *same degree k* is called a ***regular graph of degree k***, or a ***k-regular graph***.

In particular, a 3-regular graph is called a ***cubic graph***.

The graphs shown in Figures 9.35 and 9.36 are 2-regular and 4-regular graphs respectively.

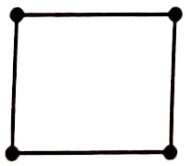


Figure 9.35

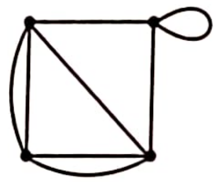


Figure 9.36

The graph shown in Figure 9.37 is a 3-regular graph (cubic graph). This particular cubic graph, which contains 10 vertices and 15 edges, is called the ***Petersen graph***.

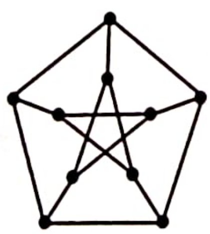


Figure 9.37

The graph shown in Figure 9.38 is a cubic graph with $8 = 2^3$ vertices. This particular graph is called the ***three-dimensional hypercube*** and is denoted by Q_3.

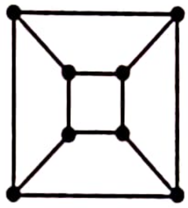


Figure 9.38

In general, for any positive integer k, a loop-free k-regular graph with 2^k vertices is called the *k-dimensional hypercube* (or ***k-cube***) and is denoted by Q_k.* (Note that the graph shown in Figure 9.35 is Q_2).

Handshaking Property

Let us refer back to the graph shown in Figure 9.33. As noted earlier, we have, in this graph,

$$\deg(v_1) = 3, \quad \deg(v_2) = 4, \quad \deg(v_3) = 4, \quad \deg(v_4) = 3.$$

Also, the graph has 7 edges. We observe that

$$\deg(v_1) + \deg(v_2) + \deg(v_3) + \deg(v_4) = 14 = 2 \times 7.$$

This observation illustrates the following important property common to all (finite) graphs.

Property : *The sum of the degrees of all the vertices in a graph is an even number; and this number is equal to twice the number of edges in the graph.*

In an alternative form, this property reads as follows:
For a graph $G = (V, E)$.

$$\sum_{v \in V} \deg(v) = 2|E|.$$

This property is obvious from the fact that while counting the degrees of vertices, each edge is counted twice (once at each end).

The aforesaid property is popularly called the *handshaking property*.† Because, it essentially states that if several people shake hands, then the total number of hands shaken must be even, because just two hands are involved in each handshake.

*The graph Q_k arises in the study of *Computer Architecture*.

†Handshaking property was the first result noted in Graph Theory. This property is also known as the ***First Theorem of Graph Theory***.

The following theorem is a direct consequence of the handshaking property.

Theorem : *In every graph, the number of vertices of odd degrees is even.*

Proof: Consider a graph with n vertices. Suppose k of these vertices are of odd degree so that the remaining $n-k$ vertices are of even degree. Denote the vertices with odd degree by $v_1, v_2, v_3, \ldots, v_k$ and the vertices with even degree by $v_{k+1}, v_{k+2}, \ldots, v_n$. Then the sum of the degrees of the vertices is

$$\sum_{i=1}^{n} \deg(v_i) = \sum_{i=1}^{k} \deg(v_i) + \sum_{i=k+1}^{n} \deg(v_i) \tag{1}$$

In view of the hand shaking property, the sum on the left hand side of the above expression is equal to twice the number of edges in the graph. As such, this sum is even. Further, the second sum in the right hand side is the sum of the degrees of vertices with even degrees. As such, this sum is also even. Therefore, the first sum in the right hand side must also be even, that is,

$$\deg(v_1) + \deg(v_2) + \cdots + \deg(v_k) = \text{even}. \tag{2}$$

But, each of $\deg(v_1), \deg(v_2), \ldots, \deg(v_k)$ is odd. Therefore, the number of terms in the left hand side of (2) must be even, that is, k is even.

This completes the proof of the theorem. •

Note: According to the above theorem, in any graph, there is an even number of vertices of odd degrees. But, it is *not true* in general that a graph must have an odd number of vertices of even degrees. Observe that the graph shown in Figure 9.33 has an even number of vertices of even degrees.

Example 1 *For the graph shown in Figure 9.39, indicate the degree of each vertex and verify the handshaking property:*

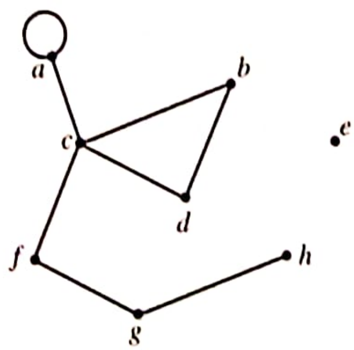


Figure 9.39

▶ By examining the graph, we find that the degrees of its vertices are as given below:

$$\deg(a) = 3, \quad \deg(b) = 2, \quad \deg(c) = 4,$$
$$\deg(d) = 2, \quad \deg(e) = 0, \quad \deg(f) = 2,$$
$$\deg(g) = 2, \quad \deg(h) = 1.$$

We note that e is an isolated vertex and h is a pendant vertex.

Further, we observe that the sum of the degrees of vertices is equal to 16. Also, the graph has 8 edges. Thus, the sum of the degrees of vertices is equal to twice the number of edges. This verifies the handshaking property for the given graph. ■

Example 2 *Can there be a graph consisting of the vertices A, B, C, D with* $\deg(A) = 2$, $\deg(B) = 3$, $\deg(C) = 2$, $\deg(D) = 2$*?*

▶ In every graph, the sum of the degrees of the vertices has to be an even number. Here, this sum is 9 which is not even. Therefore, there does not exist a graph of the given kind. ■

Example 3 *Can there be a graph with* 12 *vertices such that two of the vertices have degree* 3 *each and the remaining* 10 *vertices have degree* 4 *each?*

▶ Here, the sum of the degrees of vertices is $(3 \times 2) + (4 \times 10) = 46$. Therefore, if $m = 23$, we have $2m = 46$ and the handshaking property holds. Hence there can be a graph of the desired type (whose size is 23). ■

Example 4 *For a graph* $G = (V, E)$, *what is the largest possible value for* $|V|$ *if* $|E| = 19$ *and* $\deg(v) \geq 4$ *for all* $v \in V$*?*

▶ Since all vertices are of degree greater than or equal to 4, the sum of the degrees of vertices is greater than or equal to $4n$, where $n = |V|$ is the number of vertices. This sum is equal to twice $|E|$, by handshaking property. Therefore, $2|E| \geq 4n$, that is, $2 \times 19 \geq 4n$, or $n \leq (38/4) < 10$. Thus, the largest possible value of $|V|$ is 9. ■

Example 5 *Show that the hypercube* Q_3 *is a bipartite graph which is not a complete bipartite graph.*

▶ The hypercube Q_3 is shown in Figure 9.40, with the vertices labeled as A, B, C, D, P, Q, R, S so that, for this graph, the vertex set is $V = \{A, B, C, D, P, Q, R, S\}$.

Let

$$V_1 = \{A, C, Q, S\} \quad \text{and} \quad V_2 = \{B, D, P, R\}$$

Then, we note that $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \Phi$ and that every edge of the graph has one end vertex in the set V_1 and the other end vertex in the set V_2; further, no edge of the graph has both of its end vertices in V_1 or V_2. Hence, this graph is a bipartite graph.

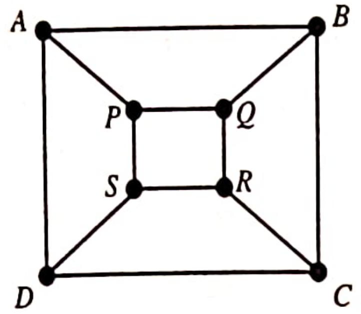


Figure 9.40

cubic: 3n
k: kn

We observe that this graph is *not* a complete bipartite graph. Because, there is no edge between every vertex in V_1 and every vertex in V_2; for example, there is no edge joining A and R. ∎

Example 6 *Prove that the k-dimensional hypercube Q_k has $k2^{k-1}$ edges.*

Determine the number of edges in Q_8.

► In the hypercube Q_k, the number of vertices is 2^k and each vertex is of degree k. Therefore, the sum of degrees of vertices of Q_k is $k \times 2^k$. By handshaking property, we should have $k \times 2^k = 2|E|$, where $|E|$ is the size of Q_k. Thus, $|E| = \frac{1}{2}(k \times 2^k) = k \times 2^{k-1}$. This means that Q_k has $k2^{k-1}$ edges.

It follows that the number of edges in Q_8 is $8 \times 2^7 = 1024$. ∎

Example 7 (a) *What is the dimension of the hypercube with 524288 edges?*

(b) *How many vertices are there in a hypercube with 4980736 edges?*

► For the k-dimensional hypercube Q_k, the number of vertices is 2^k and the number of edges is $k2^{k-1}$.

(a) If Q_k has 524288 edges, we have $k2^{k-1} = 524288$. We check that

$$524288 = 2^{19} = 2^4 \times 2^{15} = 16 \times 2^{15}.$$

Accordingly, $k2^{k-1} = 524288$ holds if $k = 16$. Thus, the dimension of the hypercube with 524288 edges is $k = 16$.

(b) We check that $4980736 = 19 \times 2^{18}$, which indicates that Q_k has 4980736 edges when $k = 19$. The number of vertices in this hypercube is $2^k = 2^{19} = 524288$. ∎

Example 8 *Determine the order $|V|$ of the graph $G = (V, E)$ in the following cases :*

(1) *G is a cubic graph with 9 edges.*

(2) *G is regular with 15 edges.*

(3) *G has 10 edges with 2 vertices of degree 4 and all other vertices of degree 3.*

► (1) Suppose the order of G is n. Since G is a cubic graph, all vertices of G have degree 3, and therefore the sum of the degrees of vertices is $3n$. Since G has 9 edges, we should have $3n = 2 \times 9$ (by the handshaking property) so that $n = 6$. Thus, the order of G is 6.

(2) Since G is regular, all vertices of G must be of the same degree, say k. If G is of order n, then the sum of the degrees of vertices is kn. Since G has 15 edges, we should have $kn = 2 \times 15$ so that $k = 30/n$. Since k has to be a positive integer, it follows that n must be a divisor of 30. Thus, the possible orders of G are $1, 2, 3, 5, 6, 10, 15$ and 30.

(3) Suppose the order of G is n. Since two vertices of G are of degree 4 and all others are of degree 3, the sum of the degrees of vertices of G is $2 \times 4 + (n - 2) \times 3$. Since G has 10 edges, we should have $2 \times 4 + (n - 2) \times 3 = 2 \times 10$. This gives $n = 6$. Thus, the order of G is 6. ■

Example 9 *Let G be a graph of order 9 such that each vertex has degree 5 or 6. Prove that at least 5 vertices have degree 6 or atleast 6 vertices have degree 5.*

► Let p be the number of vertices of G which have degree 5. Then the number of vertices of G which have degree 6 is $9 - p = q$ (say). Evidently, $0 \le p \le 9, 0 \le q \le 9$.

We note that the sum of the degrees of vertices of G is

$$(5 \times p) + (6 \times q) = 5p + 6(9 - p) = 54 - p.$$

Since this sum has to be an even number, p cannot be odd. Thus, $p = 0, 2, 4, 6,$ or 8. Consequently, the following possible cases arise :

$$\begin{aligned} p &= 0, \quad q = 9 \\ p &= 2, \quad q = 7 \\ p &= 4, \quad q = 5 \\ p &= 6, \quad q = 3 \\ p &= 8, \quad q = 1 \end{aligned}$$

We observe that in all the above possible cases *either $q \ge 5$ or $p \ge 6$*. This means that atleast 5 vertices have degree 6 or atleast 6 vertices have degree 5. ■

Example 10 *Show that there is no graph with* 28 *edges and* 12 *vertices in the following cases:*

(i) *The degree of a vertex is either* 3 *or* 4.

(ii) *The degree of a vertex is either* 3 *or* 6.

► Suppose there is a graph with 28 edges and 12 vertices, of which k vertices are of degree 3 (each). Then:

(i) If all of the remaining $(12 - k)$ vertices have degree 4, then we should have (by the handshaking property) $3k + 4(12 - k) = 2 \times 28 = 56$, or $k = -8$ which is not possible (because k has to be nonnegative).

(ii) If all of the remaining $(12-k)$ vertices have degree 6, then we should have $3k+6(12-k) = 56$, or $k = 16/3$. This is not possible (because k has to be a nonnegative integer).

Hence, in both of the two given cases, the graph of the desired type cannot exist. ∎

Example 11 (a) *If a graph with n vertices and m edges is k-regular, show that $m = kn/2$.*

(b) *Does there exist a cubic graph with* 11 *vertices?*

(c) *Does there exist a* 4*-regular graph with* (i) 15 *edges?* (ii) 10 *edges?*

►(a) If a graph G is k-regular, then the degree of every vertex is k. Therefore, if G has n vertices, then the sum of the degrees of vertices is nk. By handshaking property, this sum must be equal to $2m$ if G has m edges. Thus, $nk = 2m$, or $m = nk/2$.

(b) If there is a cubic (3-regular) graph with $n = 11$ vertices, the number of edges it should have is $m = \frac{1}{2}(3n) = \frac{1}{2}(11 \times 3)$. This is not possible, because $(33/2)$ is not a whole number. Thus, the graph of the desired type does not exist.

(c) (i) If there is a 4-regular graph with $m = 15$ edges, the number of vertices n it should have is given by $4n = 2m = 30$. This is not possible, because $30/4$ is not a whole number. Thus, the graph of the desired type does not exist.

(ii) If there is a 4-regular graph with $m = 10$ edges, the number of vertices n it should have is given by $4n = 2m = 20$. This is possible (with $n = 5$). Thus, such a graph does exist. We note that this graph is the complete graph K_5. See Figure 9.19(d) which is reproduced below,

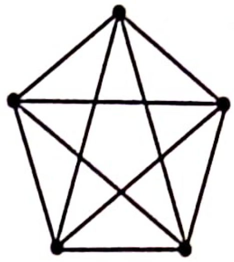


Figure 9.41

■

Example 12 (a) *If k is odd, show that the number of vertices in a k-regular graph is even.*

(b) *Show that it is not possible to have a set of seven persons such that each person in the set knows exactly three other persons in the set.*

►(a) In a k-regular graph, the degree of each vertex is k. Therefore, if such a graph has n vertices, then the sum of the degrees is nk, and this has to be an even number (by handshaking property). If k is odd, this is possible only if n is even.

(b) Let G be a graph with seven vertices, each vertex representing a person in the given set and each edge representing an acquaintance. If each person in the set knows exactly three other persons in the set, then there will be exactly three edges incident on each vertex, and the graph G will be 3-regular. This is not possible, because G has an odd number of vertices. Hence, the graph G of the desired type does not exist. This means that it is not possible to have a set of seven persons such that each person in the set knows exactly three other persons in the set. ■

Example 13 (a) *Show that in a complete graph of n vertices (namely K_n) the degree of every vertex is $(n - 1)$ and that the total number of edges is $\frac{1}{2}n(n-1)$.*

(b) *If K_n has m edges, show that $n(n + 1) = 2(n + m)$.*

►(a) Recall that a complete graph is a simple graph in which every vertex is joined with every other vertex through exactly one edge. Therefore, if there are n vertices, each vertex is joined to $(n - 1)$ vertices through exactly one edge. Thus, there occur $n - 1$ edges at every vertex. This means that the degree of every vertex is $n - 1$.

Consequently, the sum of the degrees of vertices is $n(n - 1)$. By handshaking property, this sum must be equal to $2m$, where m is the number of edges. Thus, $n(n - 1) = 2m$, or $m = \frac{1}{2}n(n-1)$. Thus, K_n has $\frac{1}{2}n(n-1)$ edges.

(b) If K_n has m edges, then we have, from what has been just proved, $m = \frac{1}{2}n(n-1)$. This gives $n + m = n + \frac{1}{2}n(n-1)$, or $2(n + m) = 2n + n^2 - n = n(n + 1)$. ■

Remark: Since the degree of every vertex in K_n is $n-1$, it follows that K_n is a $(n-1)$-regular graph.

Example 14 *Show that if a bipartite graph $G(V_1, V_2, E)$ is regular, then $|V_1| = |V_2|$.*

► Since G is simple and every edge of G has one end in V_1 and the other end in V_2,

$$\sum_{v \in V_1} \deg(v) = \sum_{v \in V_2} \deg(v) \tag{i}$$

Since G is regular, all vertices of G have the same degree, say $k > 0$. Consequently, if V_1 has r elements and V_2 has s elements,

$$\sum_{v \in V_1} \deg(v) = kr \quad \text{and} \quad \sum_{v \in V_2} \deg(v) = ks \tag{ii}$$

Using this result in (i), we get $kr = ks$, so that $r = s$. Thus, $|V_1| = |V_2|$. ■

Example 15 *Show that every simple graph of order ≥ 2 must have at least two vertices of the same degree.*

► Let G be a simple graph with n vertices. Suppose all the vertices have different degrees. Then, since every vertex must have a degree and since all such degrees must be between 0 and $n-1$, the degrees must be $0, 1, 2, 3, \ldots, n-1$. Let A be the vertex whose degree is 0 and B be the vertex whose degree is $n-1$. Then $n-1$ edges are incident on B. This means that B is joined to all other vertices by an edge and in particular to A also. Hence the degree of A is not zero. This is a contradiction. Hence all vertices of G cannot have different degrees; at least two of them must have the same degree. ■

Example 16 *Is there a simple graph with $1, 1, 3, 3, 3, 4, 6, 7$ as the degrees of its vertices?*

► Assume that there is such a graph. Since the degrees of vertices are 8 in number, the graph should have 8 vertices, say P, Q, R, S, T, U, V, W, arranged in the order of degrees as given.

Then, since there are 8 vertices and the vertex W is of degree 7, W should have an edge to all other vertices. In particular, W must have an edge to both of the vertices P and Q which are of degree 1. Then P, Q are not joined to any other vertex and in particular to the vertex V which is of degree 6. Since the graph is simple, there cannot be an edge joining V to itself. Therefore, V can be joined only to five vertices W, R, S, T, U. Then V cannot have the degree 6. This is a contradiction.

Hence there is no simple graph for which the degrees of vertices are as given. ■

Example 17 *For a graph with n vertices and m edges, if δ is the minimum and Δ is the maximum of the degrees of vertices, show that*

$$\delta \leq \frac{2m}{n} < \Delta.$$

► Let $d_1, d_2, d_3, \ldots, d_n$ be the degrees of the vertices. Then, by handshaking property, we have

$$d_1 + d_2 + d_3 + \cdots + d_n = 2m. \tag{i}$$

Since $\delta = \min(d_1, d_2, d_3, \ldots, d_n)$, we have

$$d_1 \geq \delta, \quad d_2 \geq \delta, \ldots, d_n \geq \delta.$$

Adding these n inequalities, we get

$$d_1 + d_2 + \cdots + d_n \geq n\delta. \tag{ii}$$

Similarly, since $\Delta = \max(d_1, d_2, \ldots, d_n)$, we get

$$d_1 + d_2 + \cdots + d_n \leq n\Delta. \tag{iii}$$

From (i), (ii) and (iii), we get $2m \geq n\delta$ and $2m \leq n\Delta$, so that $n\delta \leq 2m \leq n\Delta$,

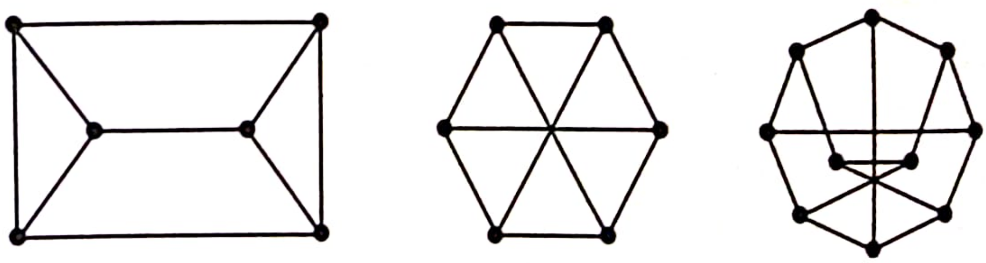
or
$$\delta \leq \frac{2m}{n} \leq \Delta$$ ■

Exercises

1. Find the degrees of all the vertices of the graph shown in Figure 9.16. Also, verify the handshaking property for this graph.

2. Verify the handshaking property for the graph shown in Figure 9.34.

3. Are the following graphs regular?



(i) (ii) (iii)

Figure 9.42

4. Draw a diagram of a graph where the degrees of the vertices are 1, 1, 1, 2, 3, 4, 5, 7.

5. Draw a diagram for the four-dimensional hypercube Q_4.

6. Show that, in a simple graph having n vertices, the degree d of every vertex satisfies the inequality $0 \leq d \leq n - 1$.

7. Consider a graph having n vertices and m edges. If p number of vertices are of degree k and the remaining vertices are of degree $k + 1$, prove that $p = (k + 1)n - 2m$.

8. For a graph $G = (V, E)$, what is the largest possible value of $|V|$ if $|E| = 35$ and $\deg v \geq 3$ for all $v \in V$?

9. How many vertices will the following graphs have, if they contain

   (i) 16 edges and all vertices of degree 4?

   (ii) 21 edges, 3 vertices of degree 4, and other vertices of degree 3?

   (iii) 12 edges, 6 vertices of degree 3, and other vertices of degree less than 3.

10. Find the fewest vertices needed to construct a complete graph with at least 1000 edges.

11. Show that there is no simple graph with four vertices such that three vertices have degree 3 and one vertex has degree 1.

12. Prove that there is no simple graph with seven vertices, one of which has degree 2, two have degree 3, three have degree 4 and the remaining vertex has degree 5.

13. Let G be a graph with exactly $n - 1$ edges. Prove that G has either a pendant vertex or an isolated vertex.

14. Prove that every cubic graph has an even number of vertices.

15. Prove that there is no simple graph for which the degree sequence is 1, 1, 1, 2, 3, 4, 5, 7.

16. Show that the maximum number of edges in a bipartite graph of order $2n$ is n^2.

17. Show that in any set of at least two persons, there are always two persons with exactly the same number of friends in the set.

18. Prove that it is impossible to have a set of nine people at a party such that each one knows exactly five of the others in the party.

19. Prove that there can be a gathering of five persons in which there are no three persons who all know each other and no three persons none of whom knows either of the other two.

Answers

1. $\deg(v_1) = 4$, $\deg(v_2) = 2$, $\deg(v_3) = 2$, $\deg(v_4) = 3$. **3.** Yes, all are 3-regular

4. See Figure below:

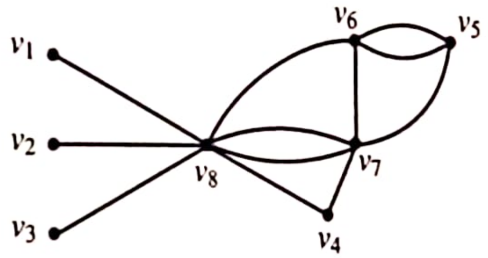


Figure 9.43

5.

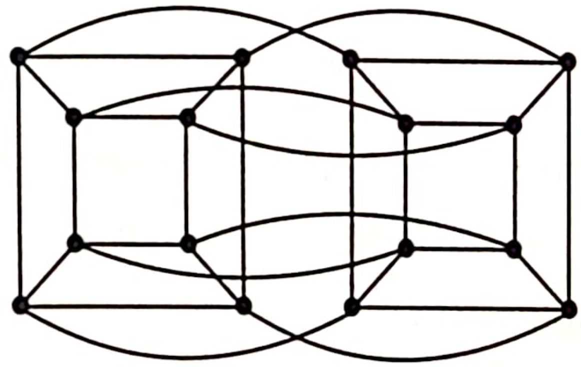


Figure 9.44

8. 23 **9.** (i) 8 (ii) 13 (iii) at least 9 **10.** 46.

9.3 Isomorphism

Consider two graphs $G = (V, E)$ and $G' = (V', E')$. Suppose there exists a function $f : V \to V'$ such that (i) f is a one-to-one correspondence*, and (ii) for all vertices A, B of G, $\{A, B\}$ is an edge of G if and only if $\{f(A), f(B)\}$ is an edge of G'. Then f is called an ***isomorphism*** between G and G', and we say that G and G' are ***isomorphic graphs***.

In other words, two graphs G and G' are said to be isomorphic (to each other) if there is a one-to-one correspondence between their vertices and between their edges such that the

*Recall the definition of one-to-one correspondence; from Section 5.3.1.

adjacency of vertices is preserved[†]. Such graphs will have the same structure; they differ only in the way their vertices and edges are labeled or only in the way they are represented geometrically. For many purposes, we regard them as essentially the same graphs.

When G and G' are isomorphic, we write $G \cong G'$.

When a vertex A of G corresponds to the vertex $A' = f(A)$ of G' under a one-to-one correspondence $f : G \to G'$, we write $A \leftrightarrow A'$. Similarly, we write $\{A, B\} \leftrightarrow \{A', B'\}$ to mean that the edge AB of G and the edge $A'B'$ of G' correspond to each other, under f.

For example, look at the two graphs shown in Figure 9.45.

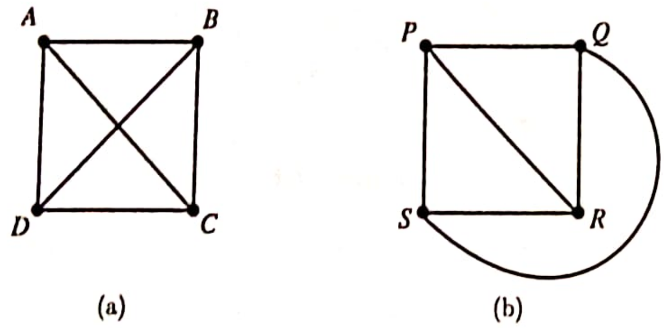


Figure 9.45

Consider the following one-to-one correspondence between the vertices of these two graphs:

$$A \leftrightarrow P, \quad B \leftrightarrow Q, \quad C \leftrightarrow R, \quad D \leftrightarrow S.$$

Under this correspondence, the edges in the two graphs correspond with each other, as indicated below:

$$\{A, B\} \leftrightarrow \{P, Q\}, \quad \{A, C\} \leftrightarrow \{P, R\}, \quad \{A, D\} \leftrightarrow \{P, S\},$$
$$\{B, C\} \leftrightarrow \{Q, R\}, \quad \{B, D\} \leftrightarrow \{Q, S\}, \quad \{C, D\} \leftrightarrow \{R, S\}$$

We check that the above-indicated one-to-one correspondence between the vertices/edges of the two graphs preserves the adjacency of the vertices. The existence of this correspondence proves that the two graphs are isomorphic. (Note that both the graphs represent the complete graph K_4)

Next, consider the two graphs shown in Figure 9.46.

Figure 9.46

[†]"adjacency of vertices is preserved" means that, for any two vertices u and v in G, if u, v are adjacent in G then the corresponding vertices u', v' in G' are also adjacent in G'.

We observe that both of these two graphs have the same number of vertices but different number of edges. Therefore, although there can exist one-to-one correspondence between the vertices, there cannot be a one-to-one correspondence between the edges. The two graphs are therefore *not* isomorphic.

From the definition of isomorphism of graphs, it follows that if two graphs are isomorphic, then they must have:

1. The same number of vertices
2. The same number of edges
3. An equal number of vertices with a given degree

These conditions are necessary but not sufficient. This means that two graphs for which these conditions hold need not be isomorphic. (See Example 8 below)

In particular, *two graphs of the same order and the same size need not be isomorphic.* To see this, consider the two graphs shown in Figure 9.47.

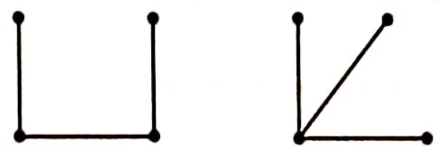


Figure 9.47

We note that both of these graphs are of order 4 and size 3. But the two graphs are *not* isomorphic. Observe that there are *two* pendant vertices in the first graph whereas there are *three* pendant vertices in the second graph. As such, under any one-to-one correspondence between the vertices and the edges of the two graphs, the adjacency of vertices is not preserved.

It is not hard to realize that every two complete graphs with the same number of vertices, n, are isomorphic. For this reason, we speak of *the* complete graph of n vertices, and all complete graphs with n vertices are denoted by K_n.

Similarly, any two complete bipartite graphs with bipartites containing r and s vertices are isomorphic. For this reason, all complete bipartite graphs with bipartites containing r and s vertices are denoted by $K_{r,s}$.

Given two graphs G and G', there is no set precedure for proving or disproving that they are isomorphic. It is only by carefully examining the nature of vertices and edges of both G and G' that one can find whether or not they are isomorphic. If G and G' are not isomorphic, it is relatively easy to find it out. If G and G' are isomorphic, the work involved in proving it is quite hard – it gets harder as the orders and sizes of G and G' get larger.

Isomorphism of Digraphs

The definition of isomorphism of graphs can be extended to digraphs in a natural way. Two digraphs D_1 and D_2 are said to be isomorphic if there is a one-to-one correspondence between their vertices and between their edges such that adjacency of vertices *along with directions* is preserved.

Example 1 *Prove that the two graphs shown below are isomorphic.*

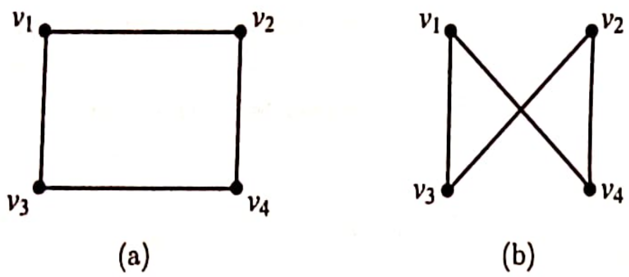


Figure 9.48

► We first observe that both graphs have four vertices and four edges. Consider the following one-to-one correspondence between the vertices of the graphs:

$$u_1 \leftrightarrow v_1, \quad u_2 \leftrightarrow v_4, \quad u_3 \leftrightarrow v_3, \quad u_4 \leftrightarrow v_2.$$

This correspondence gives the following correspondence between the edges:

$$\{u_1, u_2\} \leftrightarrow \{v_1, v_4\}, \quad \{u_1, u_3\} \leftrightarrow \{v_1, v_3\},$$
$$\{u_2, u_4\} \leftrightarrow \{v_4, v_2\}, \quad \{u_3, u_4\} \leftrightarrow \{v_3, v_2\}$$

These represent one-to-one correspondence between the edges of the two graphs under which the adjacent vertices in the first graph correspond to adjacent vertices in the second graph and vice-versa.

Accordingly, the two graphs are isomorphic. ■

Example 2 *Verify that the two graphs shown below are isomorphic.*

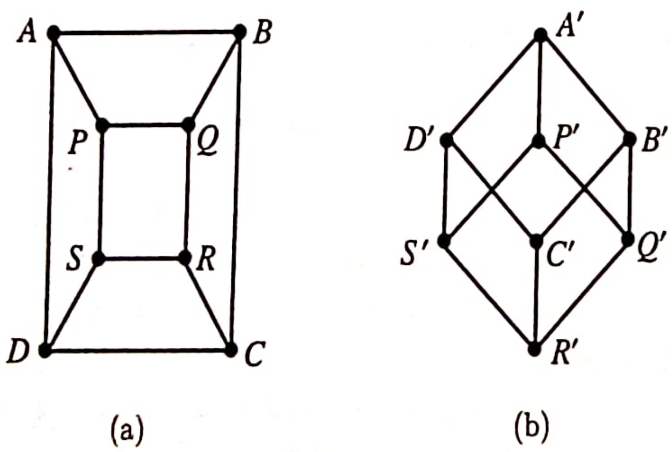


Figure 9.49

▶ Let us consider the one-to-one correspondence between the vertices of the two graphs under which the vertices A, B, C, D, P, Q, R, S of the first graph correspond to the vertices $A', B', C', D', P', Q', R', S'$ respectively of the second graph, and vice-versa. In this correspondence, the edges determined by the corresponding vertices correspond so that the adjacency of vertices is retained. As such, the two graphs are isomorphic.

We note that the first graph is the hypercube Q_3; see Figure 9.40. The second graph is just another drawing of Q_3. ■

Example 3 *Show that the following two graphs are isomorphic:*

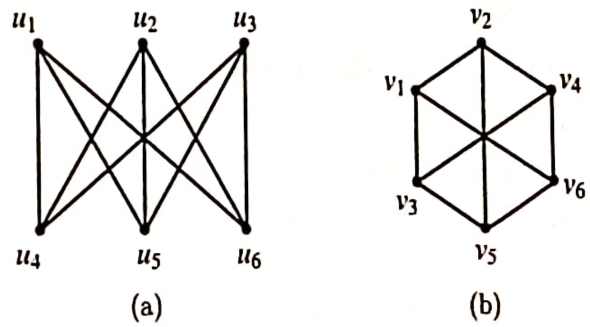


Figure 9.50

▶ We first note that both the graphs have six vertices each of degree three, and nine edges.

Bearing the edges in the two graphs in mind, consider the correspondence between the edges as shown below.

$$\{u_1,u_4\} \leftrightarrow \{v_1,v_2\}, \quad \{u_1,u_5\} \leftrightarrow \{v_1,v_3\}, \quad \{u_1,u_6\} \leftrightarrow \{v_1,v_6\},$$
$$\{u_2,u_5\} \leftrightarrow \{v_4,v_3\}, \quad \{u_2,u_4\} \leftrightarrow \{v_4,v_2\}, \quad \{u_2,u_6\} \leftrightarrow \{v_4,v_6\},$$
$$\{u_3,u_6\} \leftrightarrow \{v_5,v_6\}, \quad \{u_3,u_4\} \leftrightarrow \{v_5,v_2\}, \quad \{u_3,u_5\} \leftrightarrow \{v_5,v_3\},$$

These yield the following correspondence between the vertices:

$$u_1 \leftrightarrow v_1, \quad u_2 \leftrightarrow v_4, \quad u_3 \leftrightarrow v_5,$$
$$u_4 \leftrightarrow v_2, \quad u_5 \leftrightarrow v_3, \quad u_6 \leftrightarrow v_6.$$

We observe that the above correspondences between the edges and the vertices are one-to-one correspondences and that these preserve the adjacency of vertices. In view of the existence of these correspondences, we infer that the two graphs are isomorphic.

We note that the first graph is the complete bipartite graph $K_{3,3}$; see Figure 9.21(d). The second graph is just another drawing of $K_{3,3}$. ■

Example 4 *Show that the following two graphs are isomorphic.*

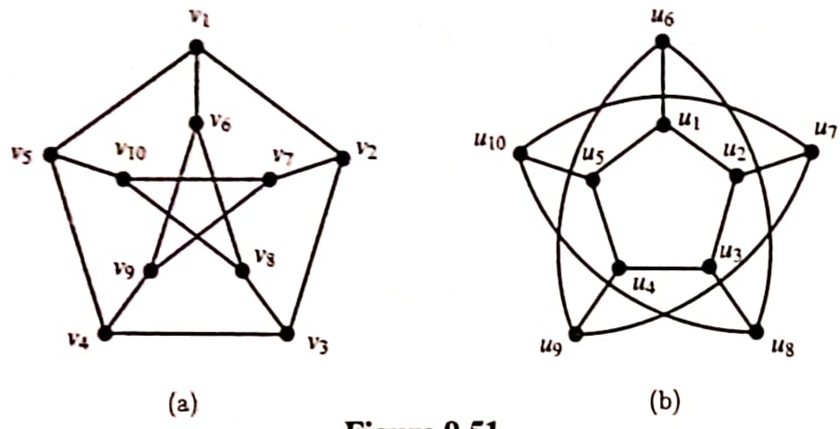


Figure 9.51

► We first note that each of the two graphs is 3-regular (cubic) and has 10 vertices. Consider the one-to-one correspondence between the vertices as shown below:

$$v_i \leftrightarrow u_i, \quad \text{for} \quad i = 1, 2, 3, \ldots, 10.$$

This correspondence has been arrived at after closely examining the structures of the two graphs.

We check that the above mentioned correspondence yields one-to-one correspondence between the edges in the two graphs with the property that adjacent vertices in the first graph correspond to the adjacent vertices in the second graph and vice-versa. The two graphs are therefore isomorphic.

We note that the first graph is the Petersen graph, see Figure 9.37. The second graph is just another drawing of the Petersen graph. ■

Example 5 *Show that the following graphs are not isomorphic.*

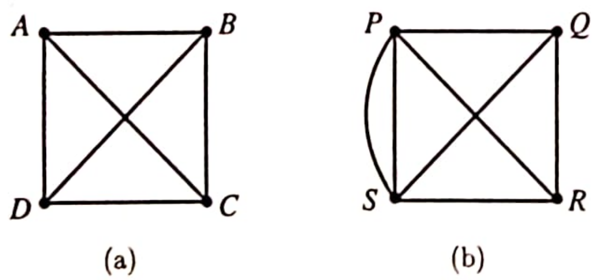
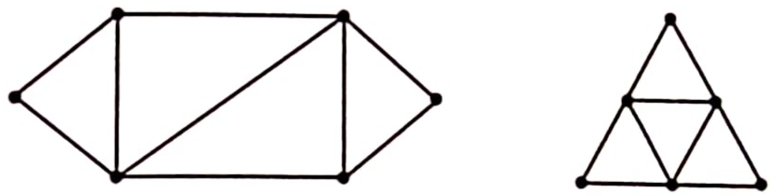


Figure 9.52

► We observe that the first graph has 4 vertices and 6 edges and the second graph has 4 vertices and 7 edges. As such, one-to-one correspondence between the edges is not possible. Hence the two graphs are *not* isomorphic. ■

Example 6 *Show that the following graphs are not isomorphic*

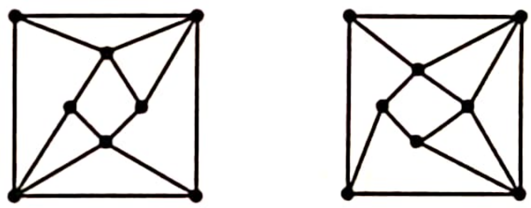


(a) (b)

Figure 9.53

► We note that each of the two graphs has 6 vertices and 9 edges. But, the first graph has 2 vertices of degree 4 whereas the second graph has 3 vertices of degree 4. Therefore, there cannot be any one-to-one correspondence between the vertices and between the edges of the two graphs which preserves the adjacency of vertices. As such, the two graphs are *not* isomorphic. ■

Example 7 *Show that the following graphs are not isomorphic.*



(a) (b)

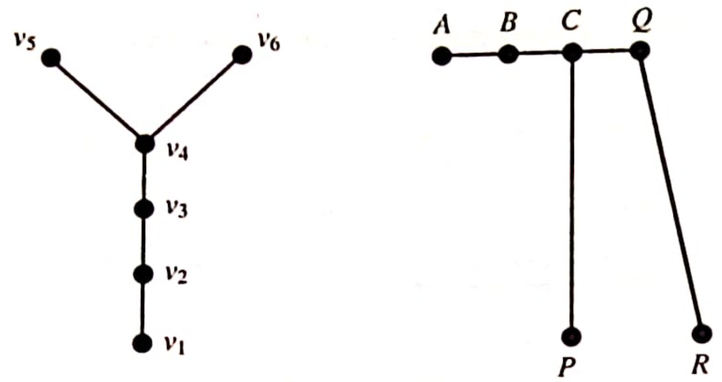
Figure 9.54

► We note that the first graph has a pair of vertices of degree 4 which are *not* adjacent whereas the second graph has a pair of vertices of degree 4 which are adjacent. (The reader is required to identify them!). Therefore the two graphs are *not* isomorphic. ■

Example 8 *Show that two graphs need not be isomorphic even if they have the same number of vertices, the same number of edges and equal number of vertices with the same degree.*

► Consider the two graphs shown in Figure 9.55:

We observe that both graphs have the same (6) number of vertices and the same (5) number of edges. Further, in each of them there are 3 vertices of degree 1 (namely, v_1, v_5, v_6 in the first graph and A, P, R in the second graph), there are 2 vertices of degree 2 (namely, v_2, v_3 in the first graph and B, Q in the second graph), and there is 1 vertex of degree 3 (namely, v_4 in the first graph and C in the second graph). Thus, the two graphs have equal number of vertices with the same degree.



(a) (b)

Figure 9.55

But, the two graphs are *not* isomorphic. Because there are 2 pendant vertices adjacent to the vertex v_4 (which is of degree 3) in the first graph but there is only one pendant vertex adjacent to the vertex C (which is of degree 3) in the second graph. As such, the adjacency of vertices cannot be preserved under a one-to-one correspondence between the vertices of the graphs. ■

Example 9 *Let $G = G(V, E)$ and $G' = G'(V', E')$ be two graphs and $f : G \to G'$ be an isomorphism. Prove the following:*

(i) *$f^{-1} : G' \to G$ is also an isomorphism.*

(ii) *For any vertex v in G,* $\deg(v)$ *in G =* $\deg(f(v))$ *in G'.*

► (i) Since $f : G \to G'$ is an isomorphism, f is a one-to-one correspondence (that is : f is one-to-one and onto). Therefore, f is invertible, and $f^{-1} : G' \to G$ is also a one-to-one correspondence.

Further, since f is an isomorphism, for all vertices a, b in G, $\{a, b\}$ is an edge in G if and only if $\{f(a), f(b)\}$ is an edge in G'. Since $a = f^{-1}\{f(a)\}$ and $b = f^{-1}\{f(b)\}$, this is equivalent to saying that $\{f(a), f(b)\}$ is an edge in G' if and only if $\{a, b\} = \{f^{-1}\{f(a)\}, f^{-1}\{f(b)\}\}$ is an edge in G.

It now follows that $f^{-1} : G' \to G$ is an isomorphism.

(ii) For any $v \in V$, let k be its degree in G. Then there exist k distinct vertics $v_1, v_2, \ldots, v_k$ such that $\{v, v_1\}, \{v, v_2\} \ldots \{v, v_k\}$ are edges in G.

Consequently, $\{f(v), f(v_1)\}, \{f(v), f(v_2)\}, \ldots \{f(v), f(v_k)\}$ are edges in G'.

Since $v_1, v_2, \ldots v_k$ are distinct and f is a one-to-one correspondence, $f(v_1), f(v_2), \ldots, f(v_k)$ are also distinct and their number is k. Thus, there are k edges in G' which are incident on $f(v)$. Hence $\deg\{f(v)\}$ in G' is k. This proves the required result. ■

Example 10 *Show that the following digraphs are isomorphic.*

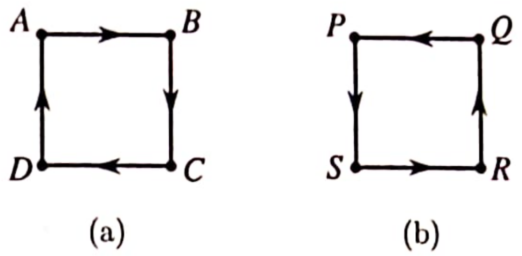


Figure 9.56

► Consider the following one-to-one correspondence between the vertices of the given digraphs:

$$A \leftrightarrow Q, \quad B \leftrightarrow P, \quad C \leftrightarrow S, \quad D \leftrightarrow R.$$

Under this correspondence, the directed edges of the two graphs correspond with each other as shown below.

$$(A, B) \leftrightarrow (Q, P), \quad (B, C) \leftrightarrow (P, S), \quad (C, D) \leftrightarrow (S, R), \quad (D, A) \leftrightarrow (R, Q)$$

Evidently, under this correspondence, the adjacency of vertices including directions of the edges is preserved.

Hence the given digraphs are isomorphic. ■

Example 11 *Show that the following digraphs are not isomorphic.*

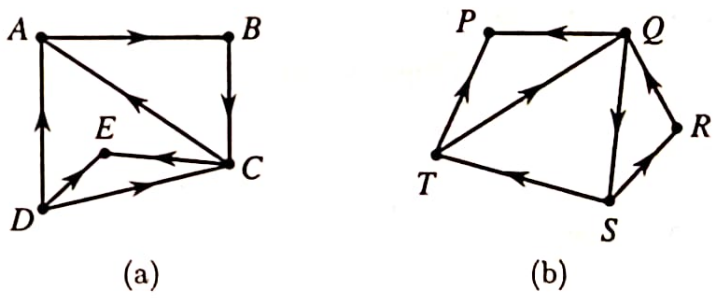


Figure 9.57

► The two digraphs have the same number of vertices (5) and the same number of directed edges (7). We observe that the vertex A of the first digraph has 1 as its out-degree and 2 as its in-degree. There is no such vertex in the second digraph. Therefore, there cannot be any one-to-one correspondence between the vertices of the two digraphs which preserves the direction of edges. The two digraphs are therefore *not* isomorphic. ■

Exercises

1. Show that the following graphs are isomorphic.

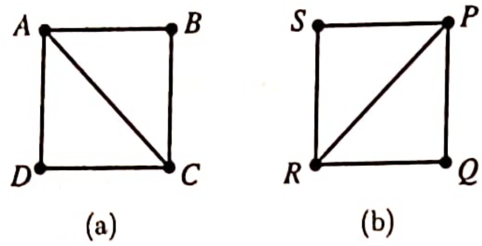


Figure 9.58

2. Show that the following graphs are isomorphic.

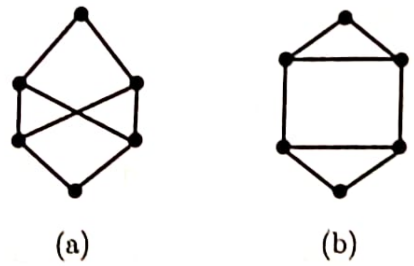


Figure 9.59

3. Show that the following graphs are isomorphic.

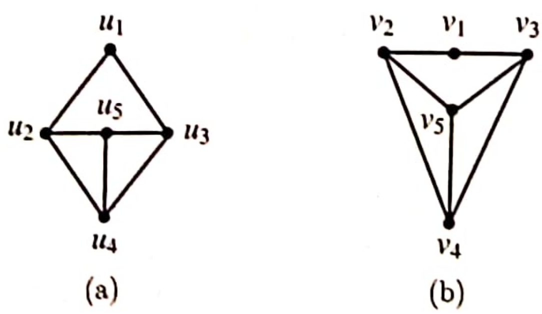
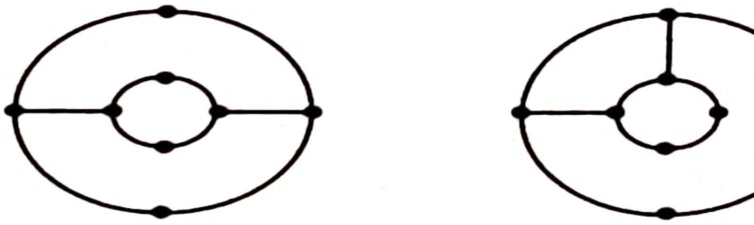


Figure 9.60

4. Show that the following graphs are not isomorphic.



(a) (b)

Figure 9.61

5. Show that the following graphs are not isomorphic.

(a) (b)

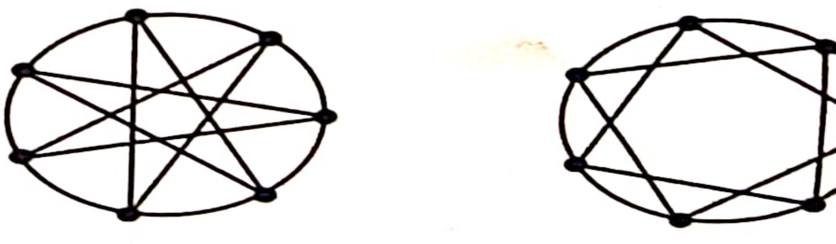
Figure 9.62

6. Show that the following graphs are not isomorphic.

(a) (b)

Figure 9.63

7. Show that the following graphs are isomorphic.



(a) (b)

Figure 9.64

8. Verify that the following graphs are isomorphic.

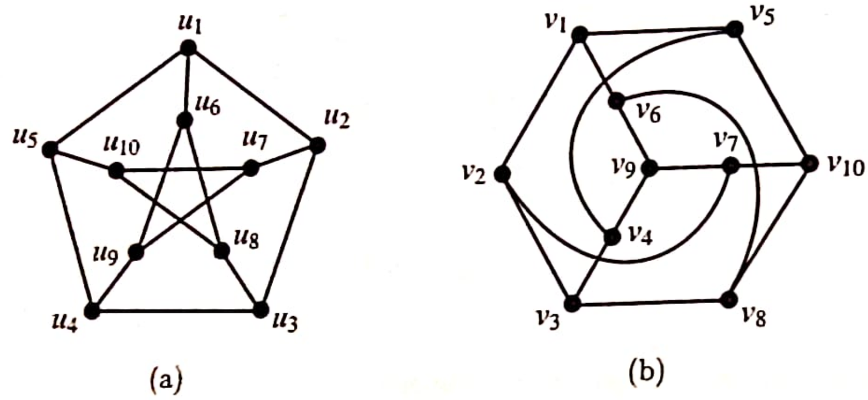


Figure 9.65

9. Show that the following digraphs are isomorphic.

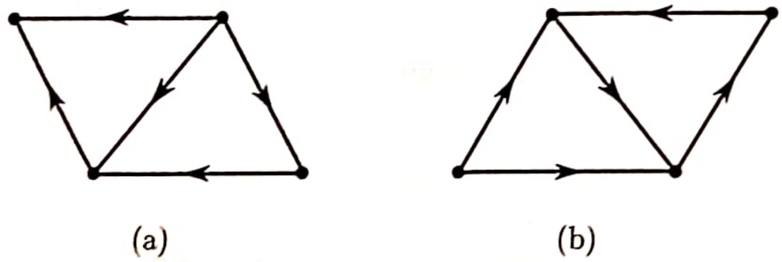


Figure 9.66

10. Show that the following digraphs are not isomorphic.

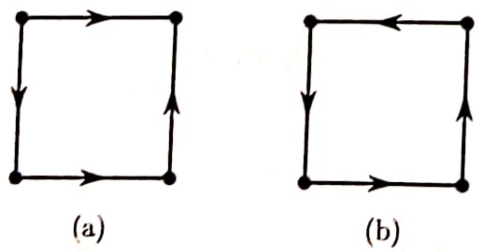


Figure 9.67

9.4 Subgraphs

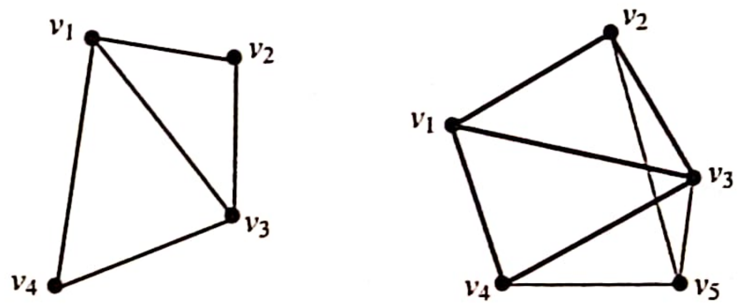
Given two graphs G and G_1, we say that G_1 is a ***subgraph*** of G if the following conditions hold.

(1) All the vertices and all the edges of G_1 are in G.

(ii) Each edge of G_1 has the same end vertices in G as in G_1.

Essentially, a subgraph is a graph which is a part of another graph. Any graph isomorphic to a subgraph of a graph G is also referred to as a subgraph of G.

Consider the two graphs G_1 and G shown in Figures 9.68(a) and 9.68(b) respectively. We observe that all vertices and all edges of the graph G_1 are in the graph G and that every edge in G_1 has the same end vertices in G as in G_1. Therefore, G_1 is a subgraph of G. In the diagram of G, the part G_1 is shown in thicker lines.



(a) : G_1 (b) : G

Figure 9.68

The following results are immediate consequences of the definition of a subgraph.

(1) Every graph is a subgraph of itself.

(2) Every simple graph of n vertices is a subgraph of the complete graph K_n.

(3) If G_1 is a subgraph of a graph G_2 and G_2 is a subgraph of a graph G, then G_1 is a subgraph of G.

(4) A single vertex in a graph G is a subgraph of G.

(5) A single edge in a graph G, together with its end vertices, is a subgraph of G.

Spanning Subgraph

Given a graph $G = (V, E)$, if there is a subgraph $G_1 = (V_1, E_1)$ of G such that $V_1 = V$, then G_1 is called a *spanning subgraph* of G.

In other words, a subgraph G_1 of a graph G is a spanning subgraph of G whenever G_1 contains all vertices of G. Thus, a graph and all its spanning subgraphs have the same vertex set. Obviously, every graph is its own spanning subgraph.

For example, for the graph shown in Figure 9.69(a), the graph shown in Figure 9.69(b) is a spanning subgraph whereas the graph shown in Figure 9.69(c) is a subgraph but not a spanning subgraph.

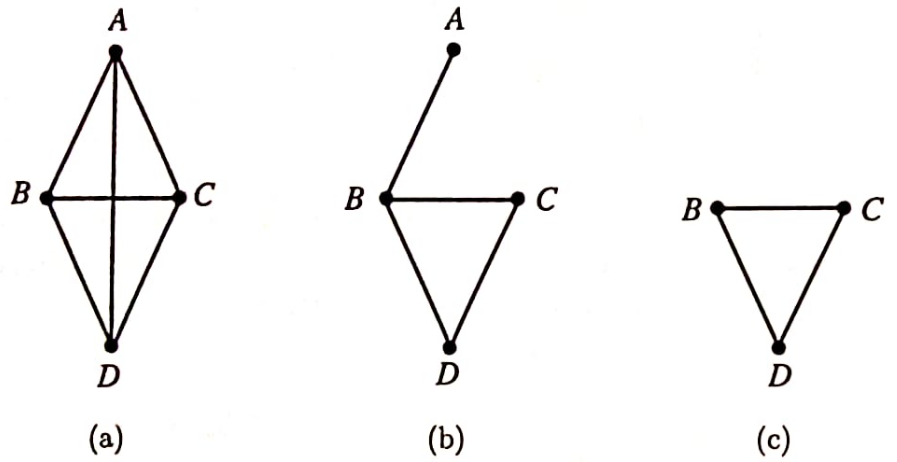


Figure 9.69

Induced Subgraph

Given a graph $G = (V, E)$, suppose there is a subgraph $G_1 = (V_1, E_1)$ of G such that every edge $\{A, B\}$ of G, where $A, B \in V_1$ is an edge of G_1 also. Then G_1 is called *an induced subgraph* of G (*inducd by* V_1) and is denoted by $< V_1 >$.

If follows that a subgraph $G_1 = (V_1, E_1)$ of a graph $G = (V, E)$ is *not* an induced subgraph of G if for *some* $A, B \in V_1$, there is an edge $\{A, B\}$ which is in G but not in G_1.

For example, for the graph shown in the Figure 9.70(a), the graph shown in Figure 9.70(b) is an induced subgraph – induced by the set of vertices $V_1 = \{v_1, v_2, v_3, v_5\}$, whereas the graph shown in Figure 9.70(c) is not an induced subgraph.

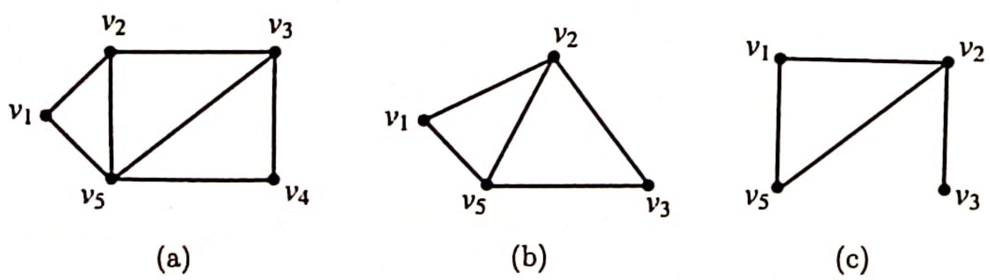


Figure 9.70

Edge-disjoint and Vertex-disjoint Subgraphs

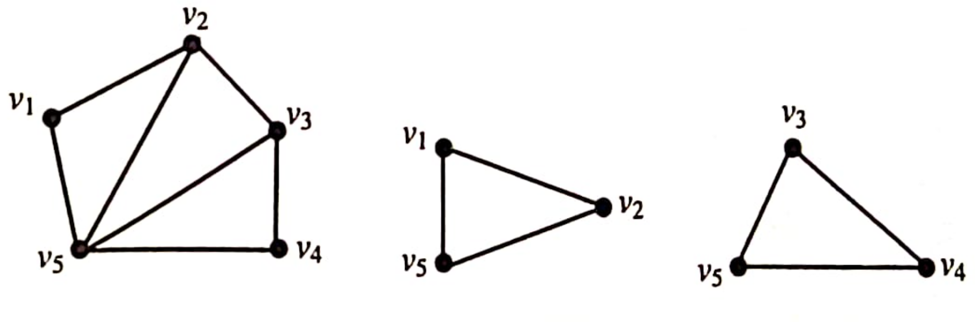
Let G be a graph and G_1 and G_2 be two subgraphs of G. Then:

(1) G_1 and G_2 are said to be *edge-disjoint* if they do not have any edge in common.

(2) G_1 and G_2 are said to be *vertex-disjoint* if they do not have any common edge *and* any common vertex.

It is to be noted that edge-disjoint subgraphs may have common vertices. Subgraphs that have no vertices in common cannot possibly have edges in common. That is, vertex-disjoint subgraphs must be edge-disjoint also but the converse is not necessarily true.

For example, for the graph shown in Figure 9.71(a), the graphs shown in Figures 9.71(b) and 9.71(c) are edge-disjoint but not vertex-disjoint subgraphs.

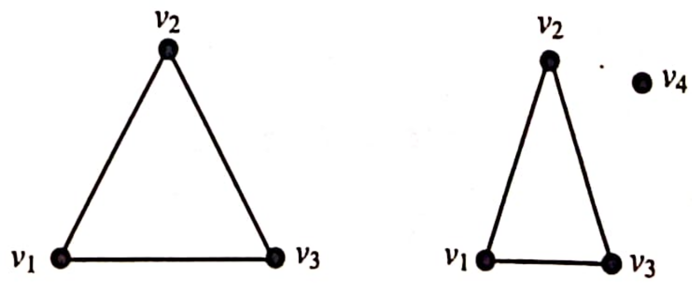


(a) (b) (c)

Figure 9.71

Example 1 *Given a graph G_1, can there exist a graph G_2 such that G_1 is a subgraph of G_2 but not a spanning subgraph of G_2 and yet G_1 and G_2 have the same size?*

► Yes. Consider a graph G_1 which contains all the vertices and all the edges of G and at least one extra isolated vertex. See Figures 9.72(a), (b) for instance.



(a) : G_1 (b) : G_2

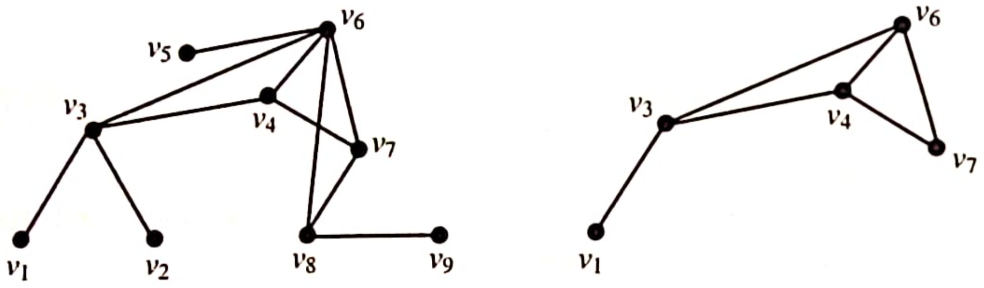
Figure 9.72

■

Example 2 *Consider the graph G shown in Figure 9.73(a).*

(a) *Verify that the graph G_1 shown in Figure 9.73(b) is an induced subgraph of G. Is this a spanning subgraph of G?*

(b) *Draw the subgraph G_2 of G induced by the set $V_2 = \{v_3, v_4, v_6, v_8, v_9\}$.*



(a) : G (b) : G_1

Figure 9.73

► (a) The vertex set of the graph G_1, namely $V_1 = \{v_1, v_3, v_4, v_6, v_7\}$, is a subset of the vertex set $V = \{v_1, v_2, \ldots, v_9\}$ of G.

Also, all the edges of G_1 are in G. Further, each edge in G_1 has the same end vertices in G as in G_1. Therefore, G_1 is a subgraph of G.

We further check that every edge $\{v_i, v_j\}$ of G where $v_i, v_j \in V_1$ is an edge of G_1. Therefore, G_1 is an induced subgraph of G. Since $V_1 \neq V$, G_1 is not a spanning subgraph of G.

(b) The subgraph $G_2 =< V_2 >$ is shown in Figure 9.73(c):

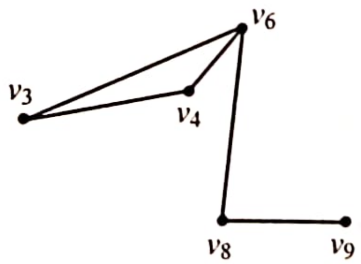


Figure 9.73: (c)

■

Example 3 *Consider the graph G shown in Figure 9.74(a). Verify that the graphs G_1 and G_2 shown in Figures 9.74(b) and 9.74(c) are induced subgraphs of G whereas the graph G_3 shown in Figure 9.74(d) is not an induced subgraph of G.*

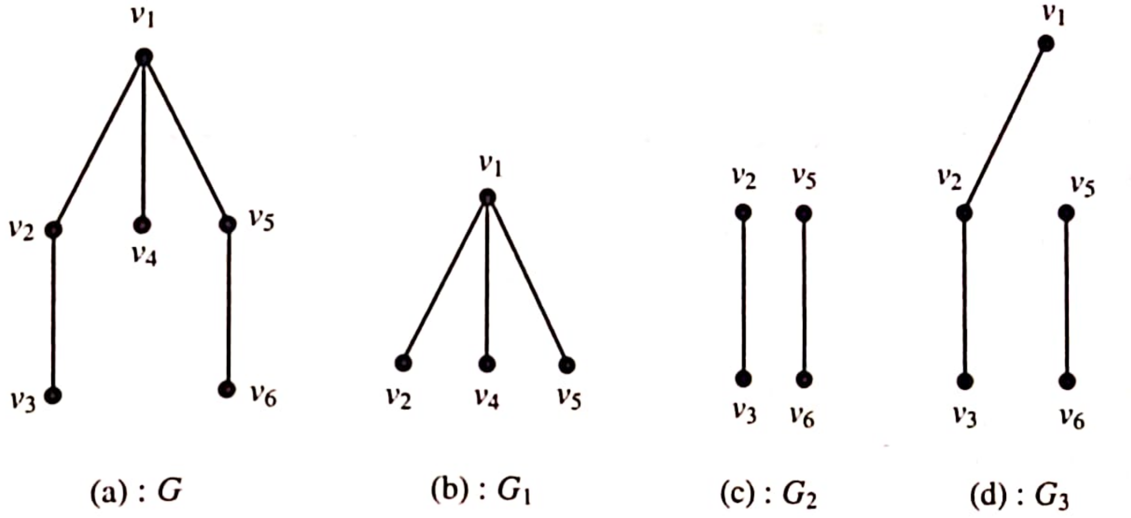


Figure 9.74

► We note that the vertex sets of G_1, G_2 and G_3 are all subsets of the vertex set of G. Further, all edges in each of G_1, G_2, G_3 have the same end vertices in G as in these. Therefore, all of G_1, G_2, G_3 are subgraphs of G.

We further check that every edge in G whose end vertices belong to G_1 is an edge in G_1. Therefore, G_1 is an induced subgraph of G. In fact, G_1 is induced by the set $\{v_1, v_2, v_4, v_5\}$.

Similarly, G_2 is an induced subgraph of G, induced by the set $\{v_2, v_3, v_5, v_6\}$.

The subgraph G_3 of G is not an induced subgraph of G. Because, for example, $\{v_1, v_5\}$ is an edge in G whose end vertices belong to G_3, but $\{v_1, v_5\}$ is not an edge in G_3. ■

Example 4 *For the graph shown in Figure 9.75, find two edge-disjoint subgraphs and two vertex-disjoint subgraphs.*

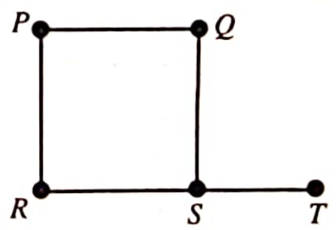


Figure 9.75

► For the given graph, two edge-disjoint subgraphs are shown in Figure 9.76(a) and two vertex-disjoint subgraphs are shown in Figure 9.76(b).

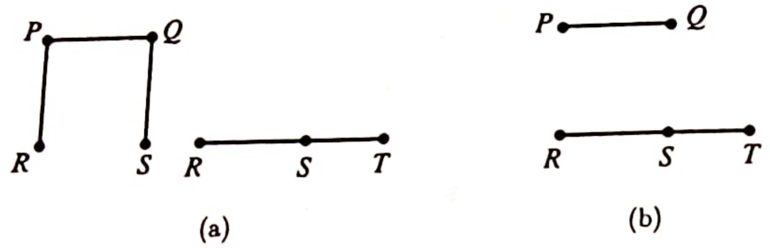


Figure 9.76

Remark: The given graph has other pairs of edge-disjoint subgraphs and vertex-disjoint subgraphs. The reader can identify these. ∎

Exercises

1. Let G be the graph shown in Figure 9.77. Verify whether $G_1 = (V_1, E_1)$ is a subgraph of G in the following cases:

   (i) $V_1 = \{P, Q, S\}$, $E_1 = \{(PQ, PS)\}$

   (ii) $V_1 = \{Q\}$, $E_1 = \Phi$, the null set

   (iii) $V_1 = \{P, Q, R\}$, $E_1 = \{PQ, QR, QS\}$,

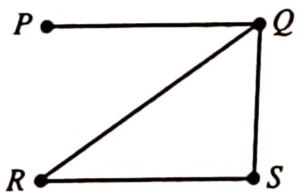
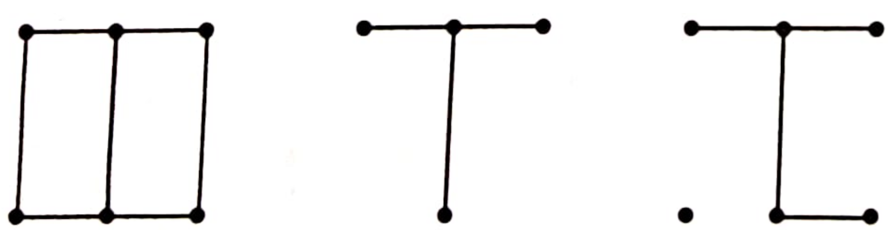


Figure 9.77

2. Three graphs G_1, G_2, G_3 are shown in Figures 9.78(a), (b), (c) respectively. Are G_2 and G_3 induced subgraphs of G_1? Are they spanning subgraphs?



(a) : G_1 (b) : G_2 (b) : G_3

Figure 9.78

3. Can a finite graph be isomorphic to one of its subgraphs (other than itself)?

Answers

1. (i) No. (ii) Yes. (iii) No.
2. G_2 is an induced subgraph of G_1, it is not a spanning subgraph.

   G_3 is not an induced subgraph of G_1, it is a spanning subgraph.
3. No.

9.5 Operations on Graphs

Consider two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.

Then the graph whose vertex set is $V_1 \cup V_2$ and the edge set is $E_1 \cup E_2$ is called the ***union*** of G_1 and G_2, it is denoted by $G_1 \cup G_2$. Thus,

$$G_1 \cup G_2 = (V_1 \cup V_2,\ E_1 \cup E_2).$$

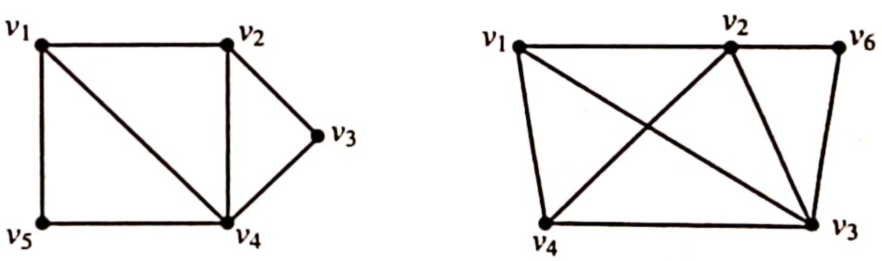
Similarly, if $V_1 \cap V_2 \neq \Phi$, the graph whose vertex set is $V_1 \cap V_2$ and the edge set is $E_1 \cap E_2$ is called the ***intersection*** of G_1 and G_2; it is denoted by $G_1 \cap G_2$. Thus,

$$G_1 \cap G_2 = (V_1 \cap V_2,\ E_1 \cap E_2). \quad \text{if} \quad V_1 \cap V_2 \neq \Phi.$$

Next, suppose we consider the graph whose vertex set is $V_1 \cup V_2$ and the edge set is $E_1 \Delta E_2$, where $E_1 \Delta E_2$ is the *symmetric difference* of E_1 and E_2*. This graph is called the ***ring sum*** of G_1 and G_2, it is denoted by $G_1 \Delta G_2$. Thus,

$$G_1 \Delta G_2 = (V_1 \cup V_2,\ E_1 \Delta E_2).$$

For the two graphs G_1 and G_2 shown in Figures 9.79(a) and (b), their union, intersection and ring sum are shown in Figures 9.80(a), (b) and (c) respectively.

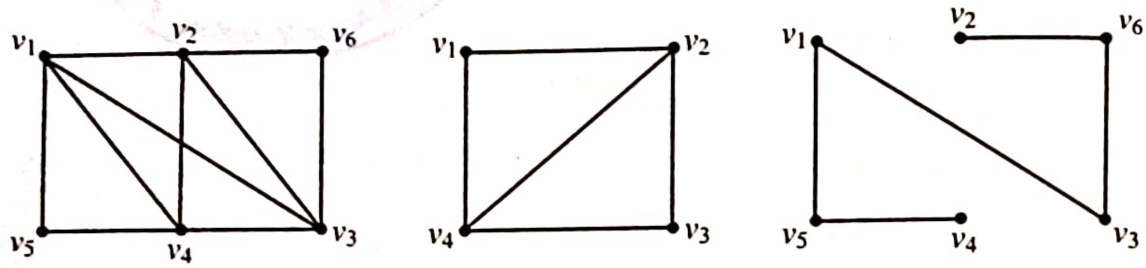


(a) : G_1 (b) : G_2

Figure 9.79

*The symmetric difference $E_1 \Delta E_2$ denotes the set of all those elements (–here, the edges) which are in E_1 or E_2 but not in both. That is:

$$E_1 \Delta E_2 = (E_1 \cup E_2) - (E_1 \cap E_2).$$



(a) : $G_1 \cup G_2$ (b) : $G_1 \cap G_2$ (c) : $G_1 \Delta G_2$

Figure 9.80

Decomposition

We say that a graph G is *decomposed* (or *partitioned*) into two subgraphs G_1 and G_2 if

$$G_1 \cup G_2 = G \quad \text{and} \quad G_1 \cap G_2 = \text{Null graph.}$$

Deletion

If v is a vertex in a graph G, then $G - v$ denotes the subgraph of G obtained by deleting v and all edges incident on v, from G. This subgraph, $G - v$, is referred to as ***vertex-deleted subgraph*** of G.

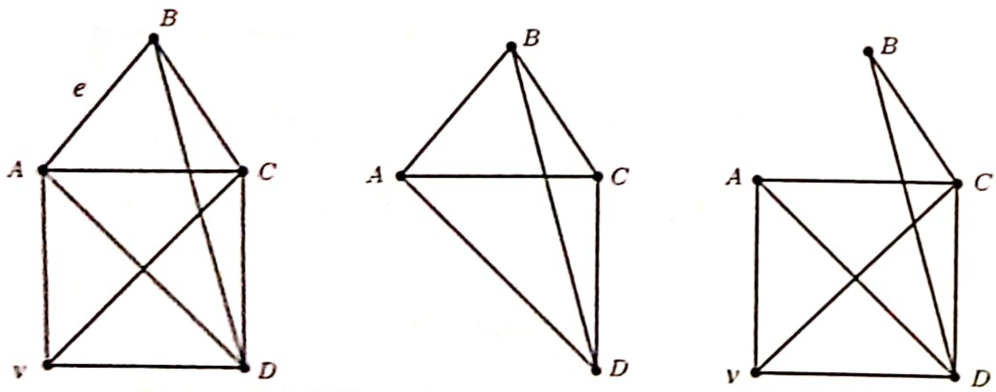
It should be noted that the deletion of a vertex always includes the deletion of all edges incident on that vertex.

Evidently, $G - v$ is the subgraph of G induced by $V_1 = V - \{v\}$.

If e is an edge in a graph G, then $G - e$ denotes the subgraph of G obtained by deleting e (but not its end vertices) from G. This subgraph, $G - e$, is referred to as ***edge-deleted subgraph*** of G.

The deletion of an edge does not alter the number of vertices. As such, an edge-deleted subgraph of a graph is a spanning subgraph of the graph.

For the graph G shown in Figure 9.81(a), the subgraphs $G - v$ and $G - e$ are shown in Figures 9.81(b) and (c) respectively.



(a) : G (b) : $G - v$ (c) : $G - e$

Figure 9.81

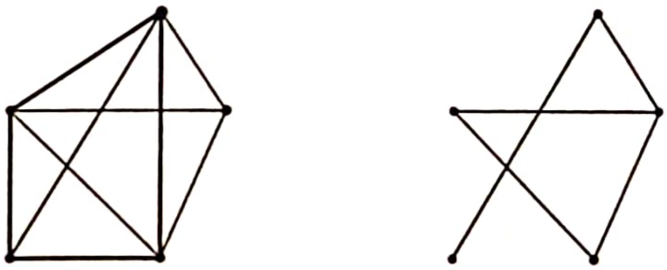
More generally, if S is any set of edges in a graph G and W is any set of vertices in G, then the subgraph of G obtained by deleting S from G is denoted by $G - S$ and the subgraph of G obtained by deleting W and all edges incident on vertices belonging to W from G is denoted by $G - W$.

Complement of a Subgraph

Given a graph G and a subgraph G_1 of G, the subgraph of G obtained by deleting from G all the edges that belong to G_1 is called the *complement* of G_1 in G; it is denoted by $G - G_1$, or $\overline{G_1}$.

In other words, if E_1 is the set of all edges of G_1, then the complement of G_1 in G is given by $\overline{G_1} = G - E_1$. We can check that $\overline{G_1} = G \Delta G_1$.

For example, consider the graph G shown in Figure 9.82(a). Let G_1 be the subgraph of G shown by thick lines in this Figure. The complement of G_1 in G, namely $\overline{G_1}$, is as shown in Figure 9.82(b).



(a) (b)

Figure 9.82

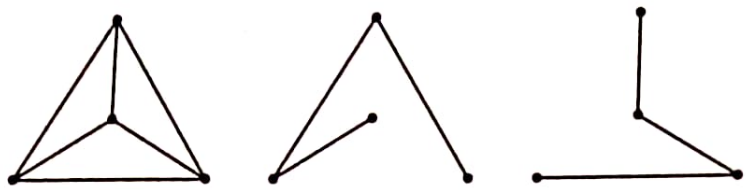
Complement of a Simple graph

Earlier, we have noted that every simple graph of order n is a subgraph of the complete graph K_n. If G is a *simple graph* of order n, then the complement of G in K_n is called the *complement* of G; it is denoted by $\overline{G}$.

Thus, the complement $\overline{G}$ of a simple graph G with n vertices is that graph which is obtained by deleting those edges in K_n which belong to G. Thus, $\overline{G} = K_n - G = K_n \Delta G$.

Evidently, K_n, G and $\overline{G}$ have the same vertex set, and two vertices are adjacent in G if and only if they are not adjacent in $\overline{G}$. Obviously, $\overline{G}$ is also a simple graph and the complement of $\overline{G}$ is G; that is $\overline{\overline{G}} = G$. It is equally obvious that the complement of K_n is the null graph of order n and vice-versa.

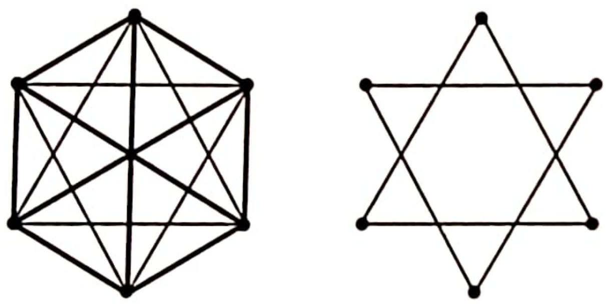
In Figure 9.83(a), the complete graph K_4 is shown. A simple graph G of order 4 is shown in Figure 9.83(b). The complement, $\overline{G}$, of G is shown in Figure 9.83(c). Observe that G, $\overline{G}$ and K_4 have the same vertices and that the edges in $\overline{G}$ are got by deleting those edges from K_4 which belong to G.



(a) : K_4 (b) : G (c) : $\overline{G}$

Figure 9.83

In Figure 9.84(a), a graph G of order 6 is shown as a subgraph of K_6, the edges of G being shown in thick lines. Its complement, $\overline{G}$, is shown in Figure 9.84(b). The graph shown in Figure 9.84(b) is known as the *David graph*.

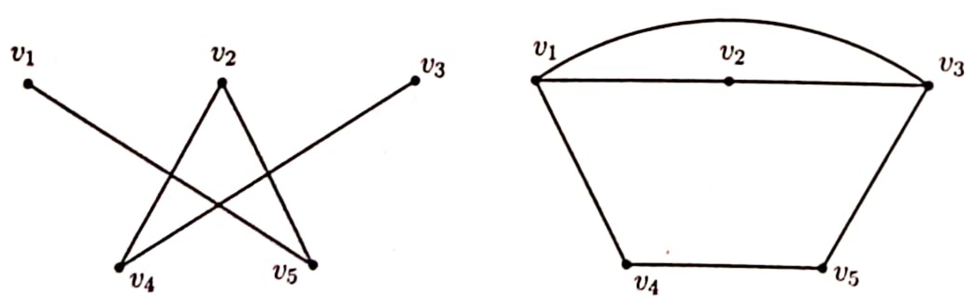


(a) (b)

Figure 9.84

Example 1 *Show that the complement of a bipartite graph need not be a bipartite graph.*

► Figure 9.85(a) shows a bipartite graph which is of order 5. The complement of this graph is shown in Figure 9.85(b), this is not a bipartite graph.



(a) (b)

Figure 9.85

■

Example 2 *Let G be a simple graph of order n. If the size of G is 56 and the size of $\overline{G}$ is 80, what is n?*

► Recall that $\overline{G} = K_n - G$. Therefore,

$$\text{size of } \overline{G} = (\text{size of } K_n) - (\text{size of } G)$$

Since size of K_n (that is : the number of edges in K_n) is $\frac{1}{2}n(n-1)$, this yields

$$80 = \tfrac{1}{2}n(n-1) - 56, \quad \text{or,} \quad n(n-1) = 160 + 112 = 272 = 17 \times 16.$$

Thus, $n = 17$. (That is, G is of order 17). ■

Example 3 *If a simple graph G of order n is isomorphic to its complement $\overline{G}$, show that n or $(n-1)$ must be a multiple of 4.*

► Since $G \approx \overline{G}$, both of G and $\overline{G}$ have the same number of edges. Also, the total number of edges in G and $\overline{G}$ taken together must be equal to the number of edges in K_n. Since K_n has $n(n-1)/2$ edges, it follows that each of G and G has $n(n-1)/4$ edges. Thus, $n(n-1)/4$ must be a positive integer. Therefore, n or $(n-1)$ must be a multiple of 4. ■

Remark: A simple graph G which is isomorphic to its complement $\overline{G}$ is called a ***self-complementary graph***. The result proved in the above Example shows that for a self-complementary graph the order n is either $4k$ or $4k+1$, where k is a positive integer, and the degree is $n(n-1)/4$.

The graph shown in Figure 9.86(a) is an Example of a self-complementary graph with 4 vertices. Its complement is shown in Figure 9.86(b)

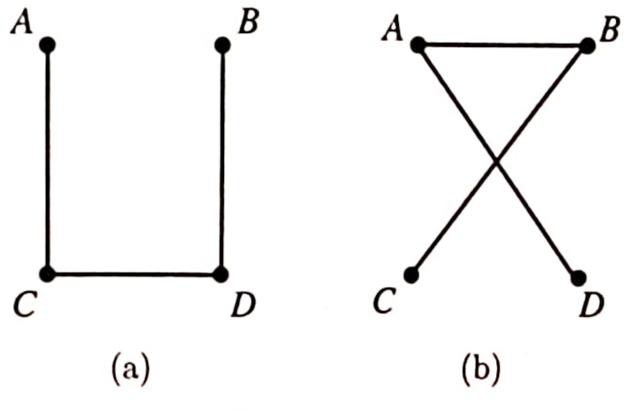


Figure 9.86

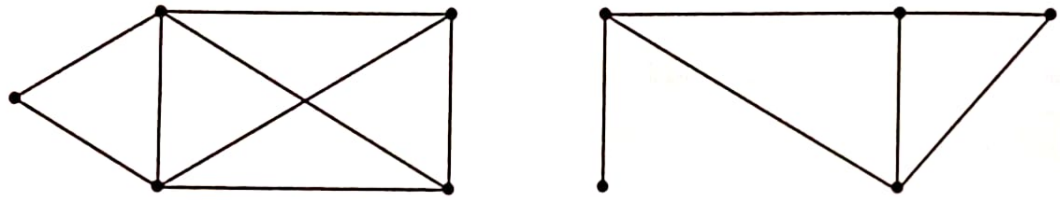
Example 4 *Prove that two simple graphs G_1 and G_2 are isomorphic if and only if their complements $\overline{G}_1$ and $\overline{G}_2$ are isomorphic.*

► Let $G_1 = (V_1, E_1)$ and $G_2(V_2, E_2)$. If G_1 and G_2 are isomorphic, then there is a one-to-one and onto function $f : V_1 \to V_2$ which preserves adjacencies of vertices. Consequently, for any

$u, v \in V_1$ if $\{u, v\} \notin E_1$ then $\{f(u), f(v)\} \notin E_2$. This means that f preserves adjacencies for $\overline{G}_1$ and $\overline{G}_2$. Therefore, f serves as an isomorphism between $\overline{G}_1$ and $\overline{G}_2$ as well. Thus, if G_1 and G_2 are isomorphic, then $\overline{G}_1$ and $\overline{G}_2$ are also isomorphic. The converse argument is analogous. ■

Exercises

1. Find the union, intersection and the ring sum of the graphs G_1 and G_2 shown below.



(a): G_1 (b): G_2

Figure 9.87

2. For the graph G shown below, find $G - v$ and $G - e$.

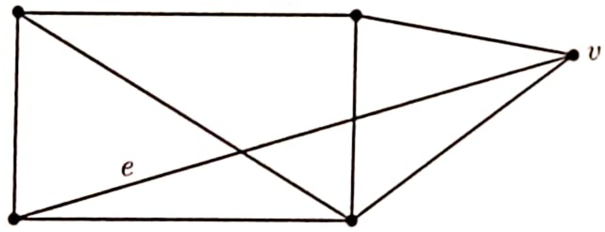


Figure 9.88

3. For graph G shown in Figure 9.89, verify that the subgraph $G - e$, where $e = \{a, d\}$ is *not* an induced subgraph of G.

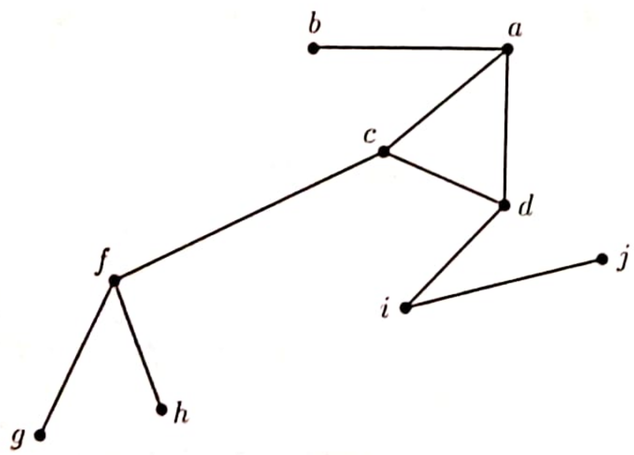


Figure 9.89

4. For the graph G and its subgraphs G_1 and G_2 shown below, find $\overline{G}_1$ and $\overline{G}_2$.

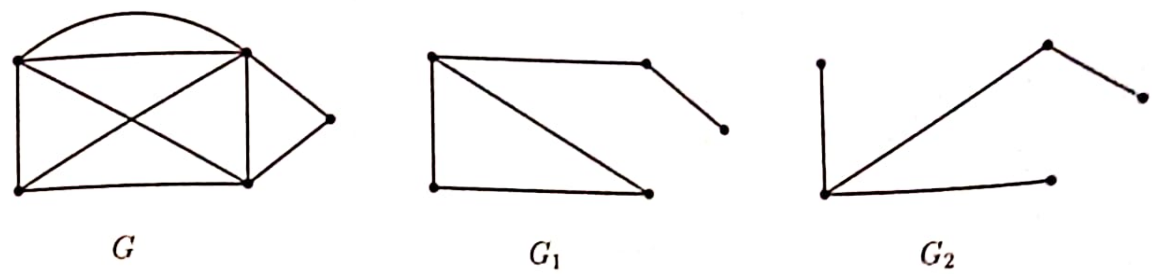


Figure 9.90

5. Find the complement of each of the following simple graphs.

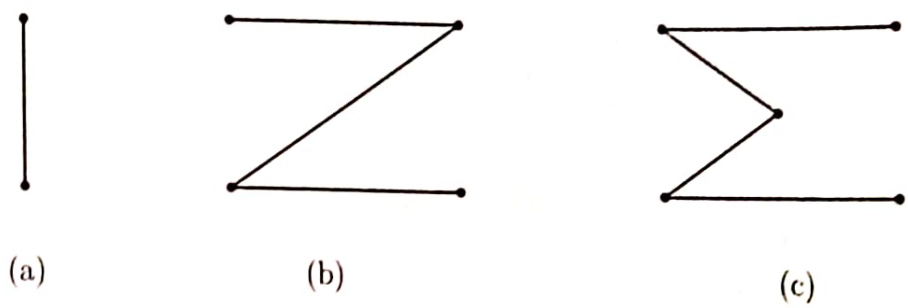


Figure 9.91

6. Draw diagrams of a self complementary graph G with five vertices and its complement $\overline{G}$.

7. Find the complement of the complete bipartite graph $K_{3,3}$.

8. Show that the complement of $K_{r,s}$ is the union of K_r and K_s.

Answers

1.

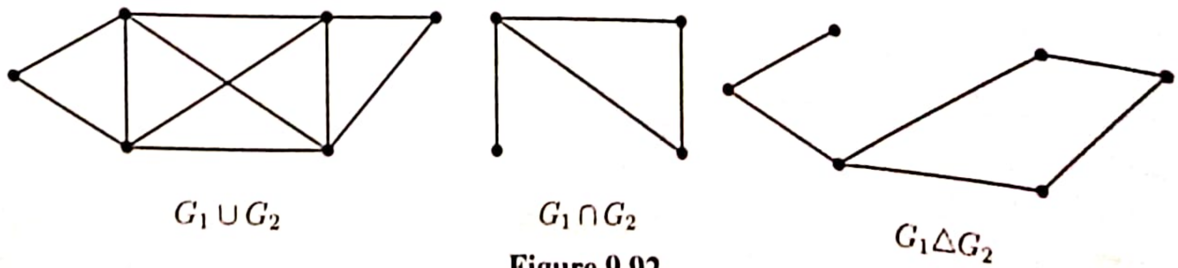


Figure 9.92

Scanned by CamScanner

2.

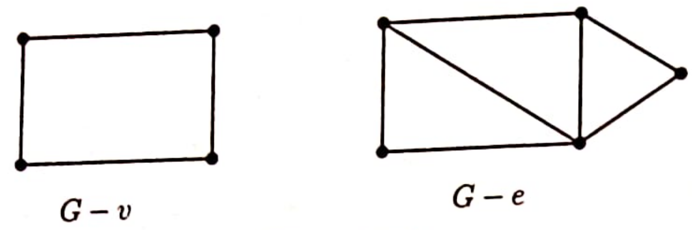


Figure 9.93

4.

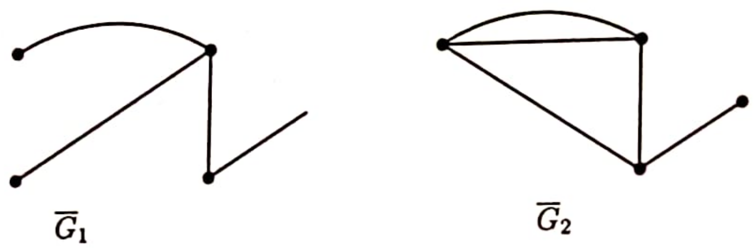


Figure 9.94

5.

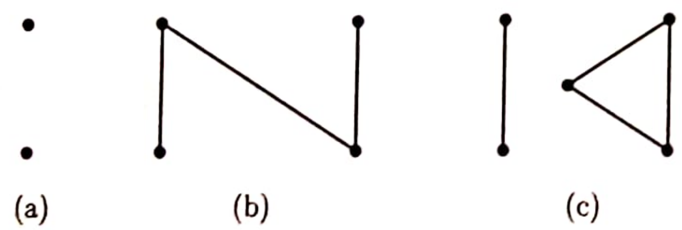


Figure 9.95

6.

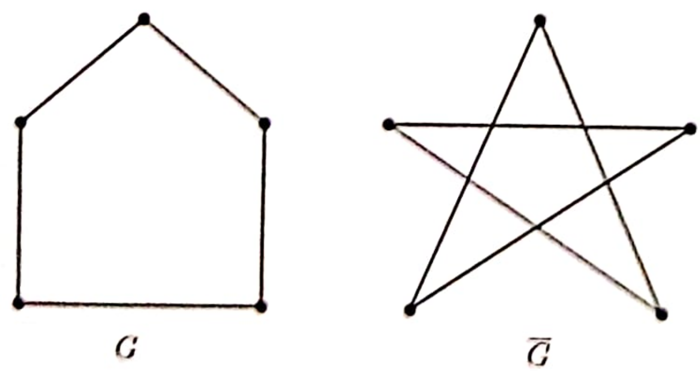


Figure 9.96

7.

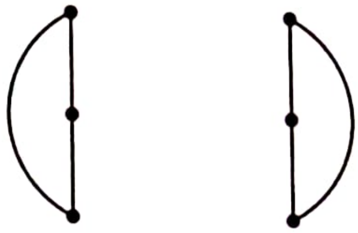


Figure 9.97

9.6 Walks and their classification

In this Section, we consider five important subgraphs of a graph, called a *walk*, a *trail*, a *circuit*, a *path* and a *cycle*. These subgraphs play a major role in studies concerned with *connected graphs* to be introduced in the next Section.

Walk

Let G be a graph having at least one edge. In G, consider a finite, alternating sequence of vertices and edges of the form

$$v_i \; e_j \; v_{i+1} \; e_{j+1} \; v_{i+2}, \ldots, e_k \; v_m$$

which begins and ends with vertices and which is such that each edge in the sequence is incident on the vertices preceding and following it in the sequence. Such a sequence is called a ***walk*** in G. In a walk, a vertex or an edge (or both) can appear more than once.

The number of edges present in a walk is called its ***length***.

For example, consider the graph shown below:

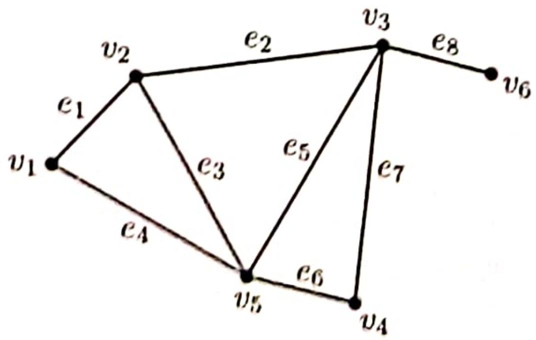


Figure 9.98

In this graph:

(i) The sequence $v_1\ e_1\ v_2\ e_2\ v_3\ e_8\ v_6$ is a walk of length 3 (because, this walk contains 3 edges: e_1, e_2, e_8). In this walk, no vertex and no edge is repeated.

(ii) The sequence $v_1\ e_4\ v_5\ e_3\ v_2\ e_2\ v_3\ e_5\ v_5\ e_6\ v_4$ is a walk of length 5. In this walk, the vertex v_5 is repeated, but no edge is repeated.

(iii) The sequence $v_1\ e_1\ v_2\ e_3\ v_5\ e_3\ v_2\ e_2\ v_3$ is a walk of length 4. In this walk, the edge e_3 is repeated and the vertex v_2 is repeated.

The vertex with which a walk begins is called the *initial vertex* (or the *origin*) of the walk and the vertex with which a walk ends is called the *final vertex* (or the *terminus*) of the walk. The initial vertex and the final vertex of a walk are together called its *terminal vertices*. The terminal vertices of a walk need not be distinct. Nonterminal vertices of a walk are called its *internal vertices*.

A walk having u as the initial vertex and v as the final vertex is called a *walk from u to v*, or briefly a u-v walk.

A walk that begins and ends at the same vertex is called a ***closed walk***. In other words, a closed walk is a walk in which the terminal vertices are coincident.

A walk which is not closed is called an ***open walk***. In other words, an open walk is a walk that begins and ends at two different vertices.

For example, in the graph shown in Figure 9.98, $v_1e_1v_2e_3v_5e_4v_1$ is a closed walk and $v_1e_1v_2e_2v_3e_5v_5$ is an open walk.

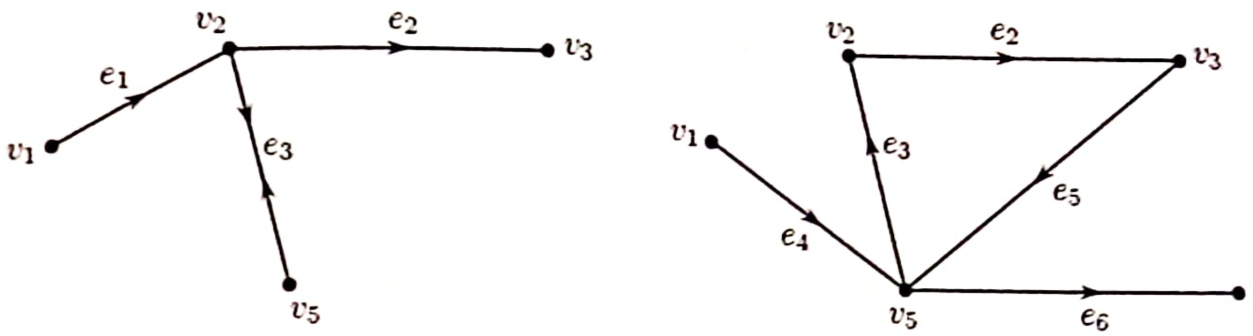
Trail and Circuit

As mentioned before, in a walk, vertices and/or edges may appear more than once. If in an open walk no edge appears more than once, then the walk is called a **trail**. A closed walk in which no edge appears more than once is called a ***circuit***.

For example, in Figure 9.98, the walk $v_1e_1v_2e_3v_5e_3v_2e_2v_3$ (shown separately in Figure 9.99(a)*) is an open walk but *not* a trail (because, in this walk, the edge e_3 is repeated) whereas the walk $v_1e_4v_5e_3v_2e_2v_3e_5v_5e_6v_4$ (shown separately in Figure 9.99(b)) is an open walk which *is* a trail.

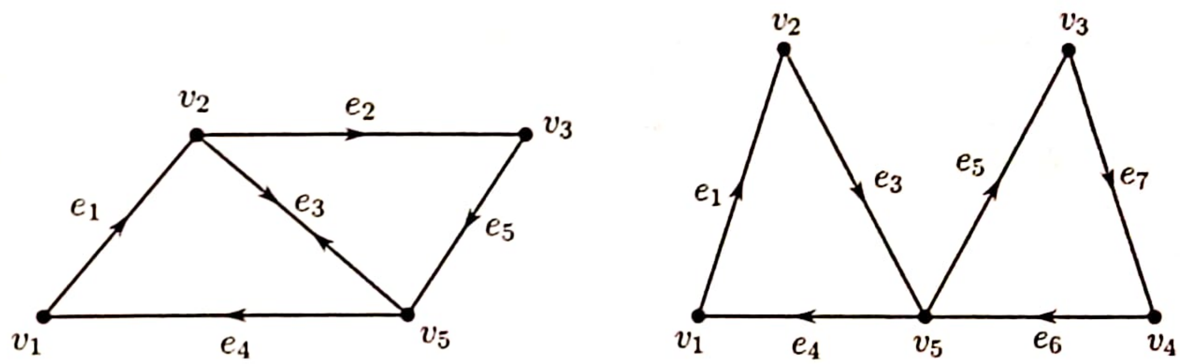
In Figure 9.98, the walk $v_1e_1v_2e_3v_5e_3v_2e_2v_3e_5v_5e_4v_1$ (shown separately in Figure 9.100(a)) is a closed walk but *not* a circuit (because e_3 is repeated) whereas the walk $v_1e_1v_2e_3v_5e_5v_3e_7v_4e_6v_5e_4v_1$ (shown separately in Figure 9.100(b)) is a closed walk which *is* a circuit.

*In Figures 9.99 to 9.102, the arrows indicate the orders in which the vertices and edges in the corresponding sequences (walks) appear.



(a): Not a trail (b): Trail

Figure 9.99



(a): Not a circuit (b): circuit

Figure 9.100

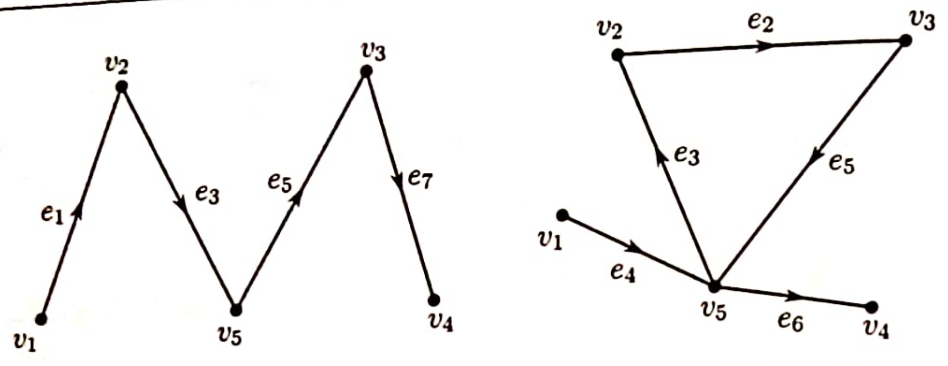
Path and Cycle

A trail in which no vertex appears more than once is called a ***path***.

A circuit in which the terminal vertex does not appear as an internal vertex and no internal vertex is repreated is called a ***cycle***.

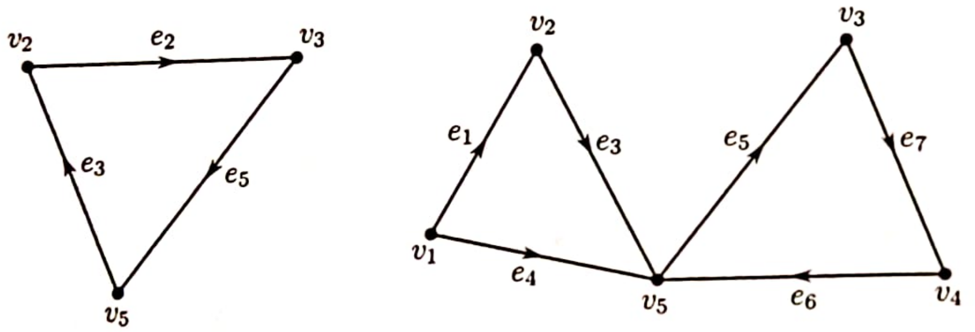
For example, in Figure 9.98, the trail $v_1e_1v_2e_3v_5e_5v_3e_7v_4$ (shown separately in Figure 9.101(a)) is a path whereas the trail $v_1e_4v_5e_3v_2e_2v_3e_5v_5e_6v_4$ (shown separately in Figure 9.101(b)) is not a path (because in this trail, v_5 appears twice).

Also, in the same Figure (i.e., in Figure 9.98), the circuit $v_2e_2v_3e_5v_5e_3v_2$ (shown separately in Figure 9.102(a)) is a cycle whereas the circuit $v_2e_1v_1e_4v_5e_5v_3e_7v_4e_6v_5e_3v_2$ (shown separately in Figure 9.102(b)) is not a cycle (because, in this circuit, v_5 appears twice).



(a): path (b): Not a path

Figure 9.101



(a): cycle (b): Not a cycle

Figure 9.102

The following facts are to be emphasised.

1. A walk can be open or closed. In a walk (closed or open), a vertex and/or an edge *can* appear more than once.

2. A trail is an open walk in which a vertex *can* appear more than once but an edge *cannot* appear more than once.

3. A circuit is a closed walk in which a vertex *can* apear more than once but an edge *cannot* appear more than once.

4. A path is an open walk in which neither a vertex nor an edge can appear more than once. Every path is a trail, but a trail need not be a path.

5. A cycle is a closed walk in which neither a vertex nor an edge can appear more than once. Every cycle is a circuit; but, a circuit need not be a cycle.

The following results are obvious:

(1) If a cycle contains only one edge, it has to be a loop.

(2) Two parallel edges (when they occur) form a cycle.

(3) In a simple graph, a cycle must have at least three edges. (A cycle formed by three edges is called a ***triangle***).

Note: While representing walks, trails, circuits, paths and cycles as sequences, only the vertices in the order of their occurrence may be indicated—omitting the edges in-between them as being understood (when there is no ambiguity).

Case of Digraphs

In the case of digraphs, the walks, trails, circuits, paths and cycles become directed walks, directed trails, directed circuits, directed paths and directed cycles. These are defined by considering sequences of vertices and edges which are consistent with the directions of edges present.

Example 1 *For the graph shown in Figure* 9.103, *indicate the nature of the following walks.*

(i) $v_1e_1v_2e_2v_3e_2v_2$ (ii) $v_4e_7v_1e_1v_2e_2v_3e_3v_4e_4v_5$ (iii) $v_1e_1v_2e_2v_3e_3v_4e_4v_5$

(iv) $v_1e_1v_2e_2v_3e_3v_4e_7v_1$ (v) $v_6e_5v_5e_4v_4e_3v_3e_2v_2e_1v_1e_7v_4e_6v_6$

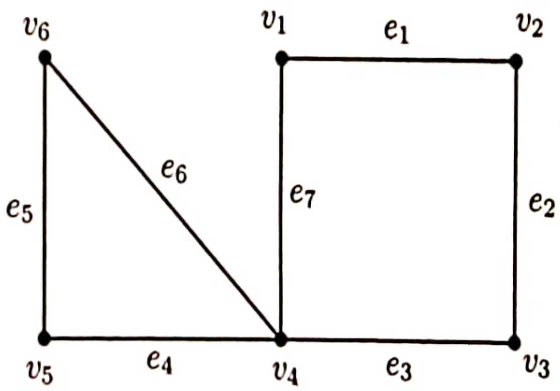


Figure 9.103

► (i) Open walk which is not a trail. (The edge e_2 is repeated).

(ii) Trail which is not a path. (The vertex v_4 is repeated).

(iii) Trail which is a path.

(iv) Closed walk which is a cycle.

Example 2 *Consider the graph shown in Figure 9.104. Find all paths from vertex A to vertex R. Also, indicate their lengths.*

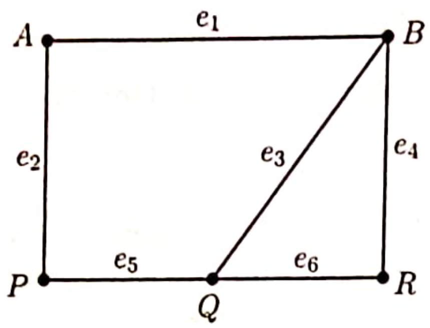


Figure 9.104

► There are four paths from A to R. These are

$$Ae_1Be_4R, \quad Ae_1Be_3Qe_6R, \quad Ae_2Pe_5Qe_6R, \quad Ae_2Pe_5Qe_3Be_4R$$

These paths contain, respectively, two, three, three and four edges. Their lengths are, therefore, two, three, three and four, respectively. ∎

Example 3 *Determine the number of different paths of length 2 in the graph shown below:*

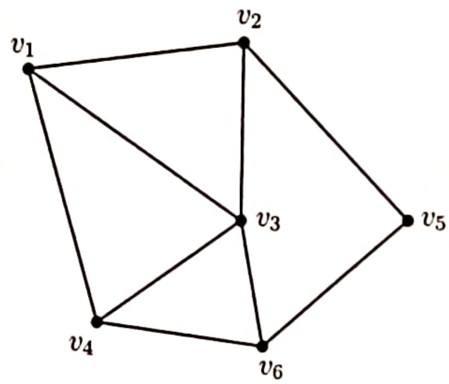


Figure 9.105

► The number of paths of length 2 that pass through the vertex v_1 is the number of pairs of edges incident on v_1. Since 3 edges are incident on v_1, this number is $3C_2 = 3$.

Similarly, the number of paths of length 2 that pass through the vertices v_2, v_3, v_4, v_5 and v_6 are, respectively,

$$3C_2 = 3, \quad 4C_2 = 6, \quad 3C_2 = 3, \quad 2C_2 = 1, \quad 3C_2 = 3.$$

Accordingly, the total number of paths of length 2 in the given graph is $3+3+6+3+1+3 = 19$. ∎

Example 4 *If G is a simple graph of order n with d_i as the degree of a vertex v_i for $i = 1, 2, \ldots n$, find the number of paths of length 2 in G.*

► Since $\deg(v_i) = d_i$, the number of edges incident on v_i is exactly d_i. Of these, every two edges give a path of length 2 which contains v_i. Therefore, there exist $C(d_i, 2)$ paths containing v_i. This is true for $i = 1, 2, \ldots n$. Therefore, the total number of paths of length 2 in G is $\sum_{i=1}^{n} C(d_i, 2)$. ■

Example 5 *If G is a simple graph in which every vertex has degree at least k, prove that G contains a path of length at least k.*

► Consider a path p in G which has a maximum number of vertices. Let u be an end vertex of p. Then every neighbour of u belongs to p. Since u has at least k neighbours (because its degree is at least k by what is given) and since G is simple, p must therefore have at least k vertices other than u. Thus, p is a path of length at least k. ■

Example 6 *Find all the cycles present in the graph shown in Figure 9.106.*

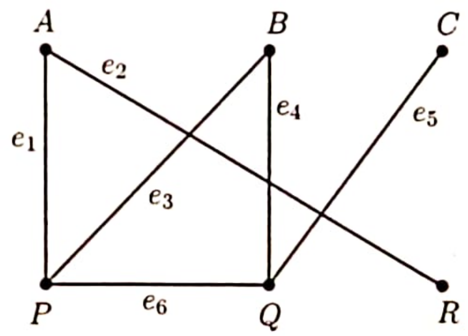


Figure 9.106

► In the given graph, there are no cycles beginning and ending with the vertices A, C and R. The cycles beginning and ending with the vertices B, P, Q are

$$Be_3Pe_6Qe_4B, \quad Pe_6Qe_4Be_3P, \quad Qe_4Be_3Pe_6Q.$$

But all of these represent one and the same cycle. Thus, there is only one cycle in the graph. ■

Example 7 *Prove the following:*

(1) *A path with n vertices is of length $n - 1$.*

(2) *If a cycle has n vertices, it has n edges.*

(3) *The degree of every vertex in a cycle is two.*

► (1) In a path, every vertex, except the last vertex, is followed by precisely one edge. Therefore, if a path has n vertices, it must have $n-1$ edges. Its length is therefore $n-1$.

(2) In a cycle, every vertex is followed by precisely one edge. Therefore, if a cycle has n vertices, it must have n edges.

(3) In a cycle, exactly two edges are incident on every vertex (– one edge through which we enter the vertex and one edge through which we leave the vertex). Therefore, the degree of every vertex in a cycle is two. ■

Example 8 *Show that, for any positive integer $k \geq 2$, there exists a simple cubic group of order $2k$.*

► Consider a set of points $v_1, v_2, \ldots v_{2k}$ and the cycle made up of the following $2k$ edges:

$$\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, v_{k+1}\}, \ldots, \ldots, \{v_{2k-1}, v_{2k}\}, \{v_{2k}, v_1\}.$$

To this cycle, let us add the following k edges:

$$\{v_1, v_{k+1}\}, \{v_2, v_{k+2}\}, \{v_3, v_{k+3}\}, \ldots, \{v_k, v_{2k}\}$$

Then the resulting graph is simple and contains $2k$ vertices and $2k + k = 3k$ edges. In this graph, exactly three edges are incident on every vertex v_i, namely the edges $\{v_{i-1}, v_i\}$ and $\{v_i, v_{i+1}\}$ which belong to the original cycle and the edge $\{v_i, v_{k+i}\}$ which has been added to the original cycle. Thus, the simple graph constructed is of order $2k$ in which the degree of every vertex is 3. This proves the existence of a graph of the desired type. ■

Example 9 *Let G be a cycle on n vertices. Prove that G is self-complementary if and only if $n = 5$.*

► Let G be a cycle of order $n = 5$ with vertices a, b, c, d, e (say) and edges $\{a, b\}$, $\{b, c\}$, $\{c, d\}$, $\{d, e\}$, $\{e, a\}$. Then $\overline{G}$ is the cycle with vertices a, b, c, d, e and edges $\{a, c\}$, $\{c, e\}$, $\{e, b\}$, $\{b, d\}$, $\{d, a\}$. It is easy to check that G and $\overline{G}$ are isomorphic.

Conversely, suppose G is a cycle on n vertices and G is self-complementary. Then the number of edges in each of G and $\overline{G}$ is $n(n-1)/4$.* Thus, we have $n(n-1)/4 = n$, or $n = 5$. ■

Example 10 *If G is a bipartite graph, show that G has no cycle of odd length.*

► Since G is bipartite, we can partition its vertex set V into two disjoint sets (bipartites) V_1 and V_2 so that each edge of G joins a vertex in V_1 and a vertex in V_2. Let $v_0 v_1 v_2 \ldots v_m v_0$ be a cycle in G, and assume (without loss of generality) that v_0 is in V_1. Then v_1 is in V_2, v_2 is in V_1, v_3 is in V_2, and so on. Thus, the vertices in the cycle belong to V_1, V_2 alternately. Since the terminal vertex of the cycle is v_0 and it is in V_1, the number of edges that belong to the cycle cannot be 3 or 5 or 7 or any odd number.

Thus, G has no cycle of odd length. ■

Remark: The converse of the result proved in the above example is also true.

*See Section 9.5, Example 3, and the Remark following that Example.

Example 11 *If G is a simple graph with no cycles, prove that G has at least one pendant vertex.*

▶ Consider a path p in G which has a maximum number of vertices. Let u be an end vertex of p. Then every neighbour of u belongs to p.† If u has at least two neighbours, say v and v', then v and v' both belong to p, and then the edges (u, v), (v, v'), (v', u) form a cycle (See Figure 9.107). This is not possible, because G has no cycles. Hence u can have only one neighbour. Accordingly, u is a pendant vertex. Thus, G has at least one pendant vertex.

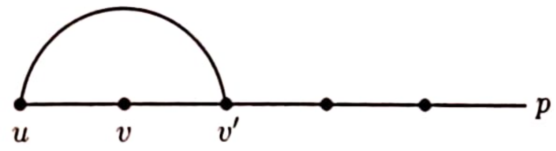


Figure 9.107 ■

Example 12 *Prove that a simple graph in which the degree of every vertex is at least two must have a cycle.*

▶ Consider a path p in G which has a maximum number of vertices. Let u be an end vertex of p. Then every neighbour of u belongs to p. Since the degree of every vertex in G is at least two, the degree of u is at least two and as such it has at least two neighbours, say v and v', both of which belong to p. Then, the edgess (u, v), (v, v') and (v', u) constitute a cycle (See Figure 9.107). Thus, G has at least one cycle. ■

Example 13 *Prove that, in a graph, there is a $u - v$ trail if and only if there is a $u - v$ path.*

▶ Since every path is a trail, if there is a $u - v$ path, it is automatic that there is a $u - v$ trail. Therefore, we need only to prove that if there is $u - v$ trail then there is a $u - v$ path.

Assume that there is a $u - v$ trail in the graph being considered. Among these trails, choose a trail of minimum length, and denote it by

$$v_0 v_1 v_2 \dots v_n \tag{i}$$

where $v_0 = u$ and $v_n = v$, and the edges between the vertices are understood. If there is only one $u - v$ trail, it will be the one with minimum length.

If, in the trail (i), no vertex is repeated then it is a path from u to v, and the proof is over. Otherwise, the trail (i) will be of the form

$$v_0 v_1 v_2 \dots v_{i-1} v_i v_{i+1} \dots v_{j-1} v_j v_{i+1} \dots v_n \tag{ii}$$

where $v_j = v_i$ for some v_i and v_j.

Consider the trail

$$v_0 v_1 v_2 \dots v_{i-1} v_i v_{j+1} \dots v_n \tag{iii}$$

†Because if a neighbour x of u does not belong to p, then we can obtain a path p' by extending p to x, and then p is no longer a path with maximum number of vertices.

which is got by skipping the vertices $v_{i+1}, v_{i+2}, \ldots v_{j-1}, v_j$ together with all edges preceding them. Evidently, this trail is shorter than the trail (ii) and we have a contradiction. Hence, the trail with minimum length has to be a path. This completes the proof. ∎

Exercises

1. For the graph shown in Figure 9.98, find the nature of the following walks (the edges in-between the vertices are understood):

   (i) $v_1v_2v_5v_3v_4v_5v_1$. (ii) $v_1v_2v_3v_5v_1$.

2. For the graph shown in Figure 9.108, find the nature of the following walks.

   (i) $BAPCB$ (ii) $PABQ$ (iii) $CBAPBQ$

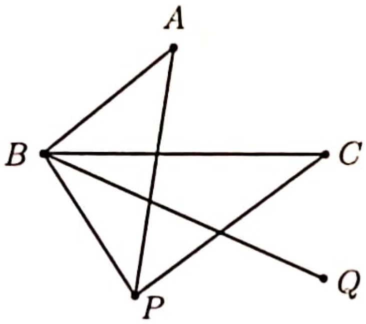


Figure 9.108

3. For the graph shown in Figure 9.109, find the nature of the following walks:

   (i) $ABEFDACDB$ (ii) $ABEFDCA$ (iii) $ACDFEBDA$ (iv) $ABDFEBDC$

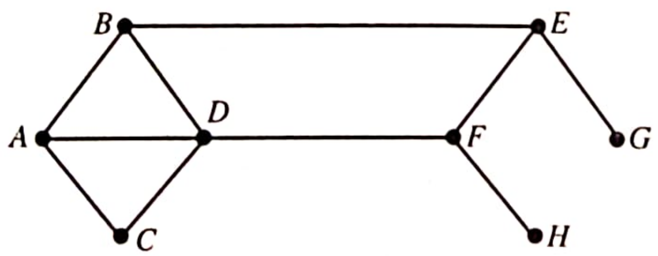


Figure 9.109

4. In the graph shown in Figure 9.110, verify that

   (i) $v_1v_2v_3v_4v_1$ is a cycle.

   (ii) $v_1v_2v_5v_3v_4v_5v_1$ is a circuit which is not a cycle.

(iii) $v_1v_2v_5v_1$ is a triangle.

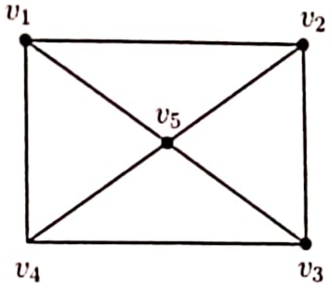


Figure 9.110

5. For the graph shown in Figure 9.111, determine (i) a walk from v_2 to v_4 which is not a trail (ii) a $v_2 - v_4$ trail which is not a path (iii) a path from v_2 to v_4 (iv) a closed walk from v_2 to v_2 which is not a circuit (v) a circuit from v_2 to v_2 which is not a cycle (vi) a cycle from v_2 to v_2, and (vii) the number of paths from v_2 to v_6.

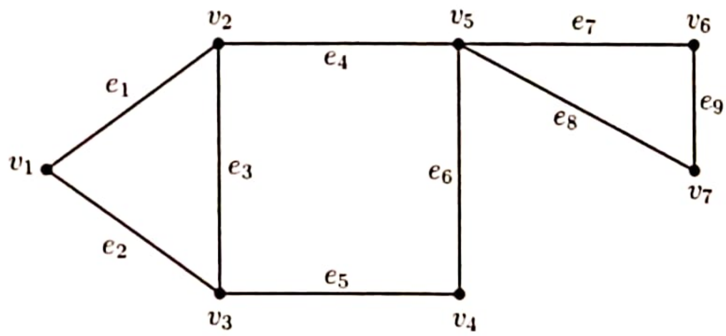


Figure 9.111

6. In the graph shown in Figure 9.112, find the number of paths from v_1 and v_8. How many of these paths have length 5?

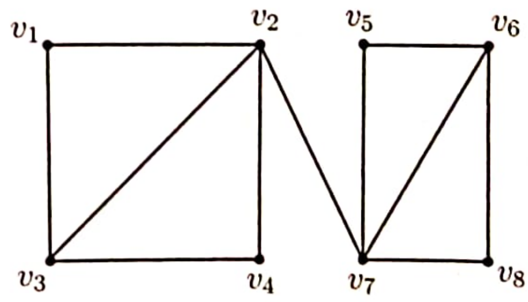


Figure 9.112

7. Verify that the complete graph K_5 has cycles with lengths 3, 4, 5.

8. Verify that in the bipartite graph $K_{3,3}$ every cycle is of length greater than or equal to four.

9. Show that in a graph with n vertices, the length of a path cannot exceed $n - 1$ and the length of a cycle cannot exceed n.

10. In a graph G, let p_1 and p_2 be two different paths between two given vertices. Prove that G has a cycle in it.

Answers

1. (i) circuit which is not a cycle. (ii) cycle.

2. (i) cycle (ii) path (iii) trail which is not a path.

3. (i) trail which is not a path (ii) cycle (iii) circuit which is not a cycle (iv) open walk which is not a trail.

5. (i) $v_2e_4v_5e_7v_6e_9v_7e_8v_5e_4v_2e_3v_3e_5v_4$ (ii) $v_2e_3v_3e_2v_1e_1v_2e_4v_5e_6v_4$

   (iii) $v_2e_3v_3e_5v_4$ (iv) $v_2e_4v_5e_7v_6e_9v_7e_8v_5e_4v_2$

   (v) $v_2e_3v_3e_5v_4e_6v_5e_7v_6e_9v_7e_8v_5e_4$ (vi) $v_2e_1v_1e_2v_3e_3v_2$ (vii) six

6. nine; three.

9.7 Connected and Disconnected Graphs

Consider a graph G of order greater than or equal to two. Two vertices in G are said to be *connected* if there is at least one *path* from one vertex to the other.*

We say that a graph G is a ***connected graph*** if every pair of distinct vertices in G are connected. Otherwise, G is called a ***disconnected graph***.

In other words, a graph G is said to be (i) connected if there is at least one path between every two distinct vertices in G, and (ii) disconnected if G has at least one pair of distinct vertices between which there is no path.

A digraph D is said to be connected or disconnected according as its underlying graph G is connected or disconnected.

Intuitively, a graph G is connected if we can reach any vertex of G from any other vertex of G by travelling along the edges, and disconnected otherwise.

*Recall that two vertices are said to be *joined* (with each other) if there is an *edge* joining them; that is if they are adjacent vertices. Two vertices that are adjacent are connected. But, *two vertices that are connected need not be adjacent*. For example, in Figure 9.104, the vertices A and R are connected but not adjacent. Thus, "joined vertices" and "connected vertices" are *not* one and the same.

For example, the graph shown in Figure 9.113(a) is connected whereas the graph shown in Figure 9.113(b) is disconnected; because, for example, there is no path from v_1 to v_4.

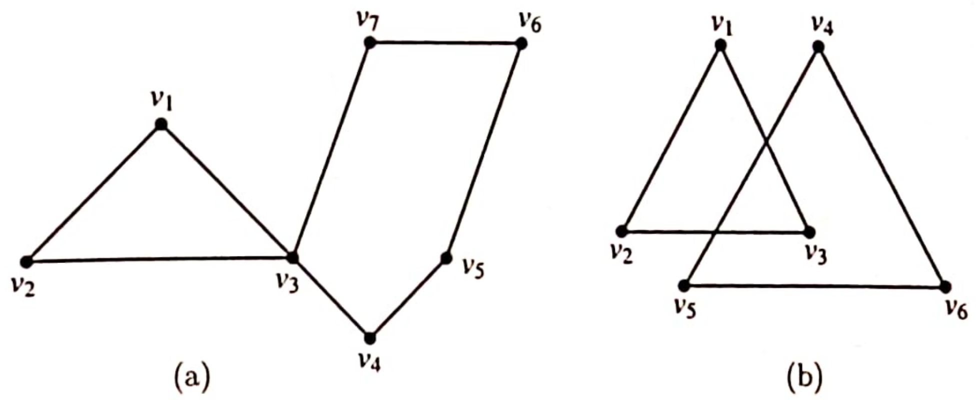


Figure 9.113

It is obvious that in a graph G all walks and, therefore, all trails, all circuits, all paths and all cycles (when they exist) are connected subgraphs of G.

It is evident that every (nontrivial) graph G consists of one or more connected graphs. Each such *connected graph* is a subgraph of G and is called a ***component*** of G.

Obviously, a connected graph has only one component and a disconnected graph has two or more components. The number of components of a graph G is denoted by $\kappa(G)$.

For example, the disconnected graph in Figure 9.113(b) has two components and so $\kappa(G) = 2$ for this graph.

It has been mentioned that the number of edges present in a walk is called the *length* of the walk. Since a path is a walk, this definition of length is applicable to paths as well. If u and v are two vertices in a connected graph, then the length of the shortest path (that is the path containing least number of edges) is called the ***distance*** between u and v.

For example, in the connected graph shown in Figure 9.113(a), the distance between the vertices v_1 and v_6 is 3 and the distance between the vertices v_3 and v_5 is 2.

The following theorems contain some useful results involving connectedness.

Theorem 1 . *If a graph has exactly two vertices of odd degree, then there must be a path connecting these vertices.*

Proof: Denote the two vertices of odd degree by v_1 and v_2. Suppose there is no path connecting these. Then the graph is disconnected, and v_1 and v_2 belong to two different components, say H_1 and H_2. Consequently, each of H_1 and H_2 contains only one vertex of odd degree. This is not possible, because H_1 and H_2 are graphs and in a graph the number of vertices of odd degrees is always even *. Hence, there must be a path connecting v_1 and v_2. This completes the proof. •

*See Theorem following the handshaking property, in Section 9.2.1.

Theorem 2 . *A connected graph with n vertices has at least n – 1 edges.*

Proof: Since the graph is connected, $n \geq 2$. If m denotes the number of edges, we have to prove that $m \geq n - 1$, for every positive integer $n \geq 2$. We employ the method of induction to prove this result.

Suppose $n = 2$. Then there are exactly two vertices in the graph and since the graph is connected, there must be at least one edge joining there vertices. Thus, now, $m \geq 1 = (2-1) = (n-1)$. This verifies the required result for $n = 2$.

Assume that the result $m \geq n - 1$ holds for all connected graphs with $n = k$ number of vertices, where k is a positive integer ≥ 2.

Now, consider a connected graph, say G_{k+1}, with $k + 1$ vertices. Choose a vertex v of this graph and consider the graph G_k obtained by deleting an edge from G_{k+1} for which v is an end vertex. Then, G_k is a connected graph with k vertices. Let m_k be the number of edges in G_k. Then from the assumption made in the preceding paragraph we have $m_k \geq k - 1$. Consequently,

$$m_k + 1 \geq (k+1) - 1.$$

But, $m_k + 1$ is the number of edges in G_{k+1} and $k + 1$ is the number of vertices in G_{k+1}. Thus, the result $m \geq n - 1$ holds for $n = k + 1$ when it holds for $n = k \geq 2$. Hence, by induction, the result holds for all integers $n \geq 2$. This completes the proof. •

Theorem 3 . *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 verex is in V_1 and the other is in V_2.*

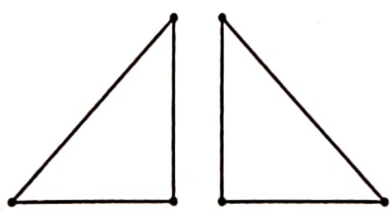
Proof: First, suppose that G is a disconnected graph. Consider a vertex v in G. Let V_1 be the set of all vertices in G that are connected to v. Since G is disconnected, V_1 does not include all vertices of G. This means that V_1 is a proper subset of V. Let $V_2 = V - V_1$. Then $V_1 \cap V_2 = \Phi$, $V = V_1 \cup V_2$ and no vertex in V_1 is connected to any vertex in V_2. Hence, V_1 and V_2 form a partition of V of the desired type.

Conversely, suppose two subsets V_1 and V_2 of V form a partition of V of the desired type. Consider two arbitrary vertices v and u in G, such that $v \in V_1$ and $u \in V_2$. Then there exists no path between v and u. Hence G is not connected.

This completes the proof of the theorem. •

Example 1 *Let G be a graph with n vertices where n is even and > 2. If the degree of every vertex in G is $\frac{1}{2}(n-2)$, disprove that G is connected.*

► Consider the disconnected graph shown below:



In this graph the number of vertices is $n = 6$ which is even and greater than 2, and the degree of every vertex is $2 = (6 - 2)/2 = (n - 2)/2$.

Thus, the graph considered meets the given conditions and is disconnected. This counter example disproves that G is connected. ∎

Example 2 *If G is a simple graph with n vertices in which the degree of every vertex is at least $(n - 1)/2$, prove that G is connected.*

▶ Take any two vertices u and v of G. Then they are either adjacent or not adjacent. If they are adjacent, then G is connected. Otherwise, each has at least $(n - 1)/2$ neighbours, because the degree of every vertex is at least $(n - 1)/2$. Therefore, u and v taken together have at least $n - 1$ neighbours. But, since G has a total of n vertices, the total number of neighbours which u and v together can have is only $n - 2$. Therefore, at least one vertex, say x, is a neighbour of both u and v. Hence, there is an edge between u and x and there is an edge between x and v. Thus, there is a path between u and v. As such, G must be connected. ∎

Example 3 *Prove that a connected graph G remains connected after removing an edge e from G if and only if e is a part of some cycle in G.*

▶ Suppose e is a part of some cycle C in G. Then the end vertices of e (say, A and B) are joined by at least two paths, one of which is e and the other $C - e$. (See Figure 9.115 wherein e is shown by thick arc). Hence the removal of e from G will not affect the connectivity of G; because even after the removal of e the end vertices of e (i.e., A and B) remain connected (through the path $C - e$).

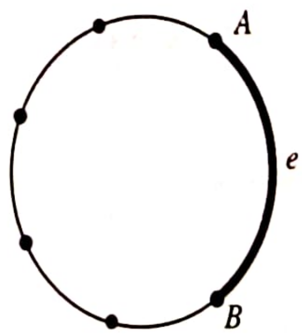
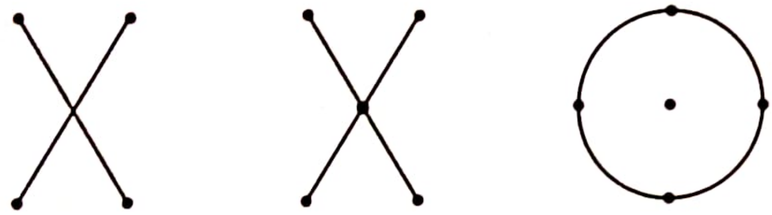


Figure 9.115

Conversely, suppose e is not a part of any cycle in G. Then the end vertices of e are connected by at most one path. Hence the removal of e from G disconnects these end points. This means that $G - e$ is a disconnected graph. Thus, if e is not a part of any cycle in G then

Exercises

1. Indicate which of the following graphs are connected.



(a) (b) (c)

Figure 9.116

2. Find the distance between the vertex v_1 and the vertices v_3, v_5, v_6 and v_{11} in the following connected graph.

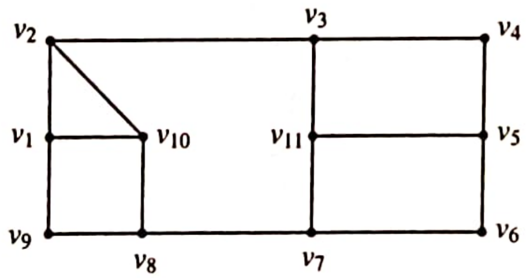


Figure 9.117

3. If G is a simple graph with n vertices and k components, prove that G has at least $n - k$ number of edges.

4. Prove that every graph with n vertices and m edges has at least $n - m$ components.

5. Prove that a connected graph of order n contains exactly one cycle if and only if its size is also n.

6. Let G be a simple graph. Show that if G is not connected then its complement $\overline{G}$ is connected.

7. Prove that if a connected graph G is decomposed into two subgraphs H_1 and H_2, there must be at least one vertex common to H_1 and H_2.

8. In a graph G, if the intersection of two paths is a disconnected subgraph, show that the union of the two paths contains at least one cycle.

9. Let G be a graph with 15 vertices and 4 components. Prove that at least one component of G has at least 4 vertices.

10. Show that if G is a connected graph in which every vertex has degree either 1 or 0 then G is either a path or a cycle.

11. Suppose the graphs G_1 and G_2 are isomorphic. Prove that if G_1 is connected then G_2 is also connected.

12. Prove that any two simple connected graphs with n vertices, all of degree two, are isomorphic.

Answers

1. Only the second graph is connected

2. $v_3 : 2, \quad v_5 : 4, \quad v_6 : 4, \quad v_{11} : 3.$

9.8 Euler circuits and Euler trails

Consider a connected graph G. If there is a *circuit* in G that contains *all the edges* of G, then that circuit is called an ***Euler circuit*** (or *Eulerian line, or Euler tour*) in G. If there is a *trail* in G that contains *all the edges* of G, then that trail is called an ***Euler trail*** (or *unicursal line*) in G.

Recall that in a trail and a circuit no edge can appear more than once but a vertex can appear more than once. This property is carried to Euler trails and Eular circuits also.

Since Euler circuits and Euler trails include all edges, they automatically should include all vertices as well.

A connected graph that contains an Euler circuit is called an ***Euler graph*** (or *Eulerian graph*).

A connected graph that contains an Euler trail is called a ***semi-Euler graph*** (or a *semi-Eulerian graph* or *unicursal graph*).

For example, in the graph shown in Figure 9.118 the closed walk

$$Pe_1Qe_2Re_3Pe_4Se_5Re_6Te_7P$$

is an Euler circuit. Therefore, this graph is an Euler graph.

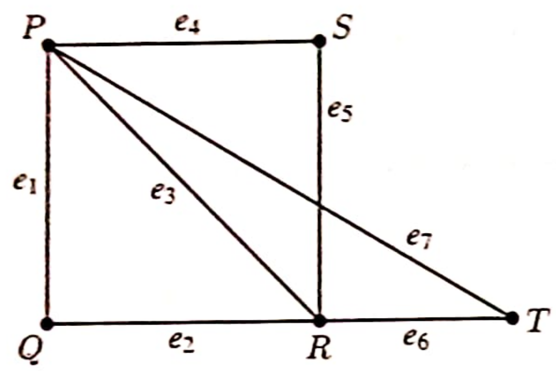


Figure 9.118

Consider the graph shown in Figure 9.119. We observe that, in this graph, every sequence of edges which starts and ends with the same vertex and which includes all edges will contain at least one repeated edge. Thus, this graph has no Eular circuits. Hence this graph is *not* an Euler graph.

It may be seen that in the graph in Figure 9.119 the trail $Ae_1Be_2De_3Ce_4Ae_5D$ is an Euler trail. This graph is therefore a semi-Euler graph.

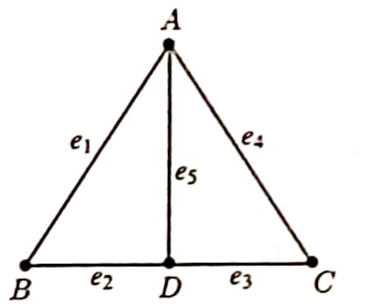


Figure 9.119

The following Theorems contain some basic properties of Euler graphs.

Theorem 1. *A connected graph G has an Euler circuit (that is, G is an Euler graph) if and only if all vertices of G are of even degree.*

Theorem 2. *A connected graph G has an Euler circuit (that is, G is an Euler graph) if and only if G can be decomposed into edge-disjoint cycles.*

Example 1 *Find all positive integers $n (\geq 2)$ for which the complete graph K_n contains an Euler circuit. For which n does K_n have an Euler trail but not an Euler circuit?*

► For $n = 2$, the graph K_n contains exactly one edge. This edge together with its end vertices constitutes an Euler trail. In this case, K_n cannot have an Euler circuit. For $n \geq 3$, K_n contains an Euler circuit if and only if $n - 1$ (which is the degree of every vertex in K_n) is even; that is if and only if n is odd. ■

Example 2 (a) *Does there exist an Euler graph with even number of vertices and odd number of edges?*

(b) *Does there exist an Euler graph with odd number of vertices and even number of edges?*

► (a) Yes. Suppose C is a circuit with even number of vertices. Let v be one of these vertices. Consider a circuit C' with odd number of vertices passing through v such that C and C' have no edge in common. The circuit q that consists of the edges of C and C' is an Euler graph of the desired type.

(b) Yes. In (a), suppose C and C' are circuits with odd number of vertices. Then q is an Euler graph of the desired type. ■

Example 3 *Show that a connected graph with exactly two vertices of odd degree has an Euler trail.*

► Let A and B be the only two vertices of odd degree in a connected graph G. Join these vertices by an edge e (even if there is already an edge between them). Then A and B become vertices of even degree. Since all other vertices in G are of even degree, the graph $G_1 = G \cup e$ is connected and has all vertices of even degree. Therefore, G_1 contains as Euler circuit which must include e. The trail got by deleting e from this Euler circuit is an Euler trail in G. ■

Exercises

1. Show that the graph shown below is an Euler graph.

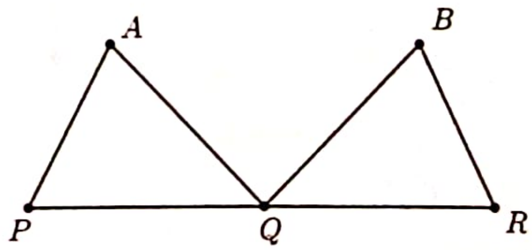


Figure 9.120

2. Find an Euler circuit in the graph shown below.

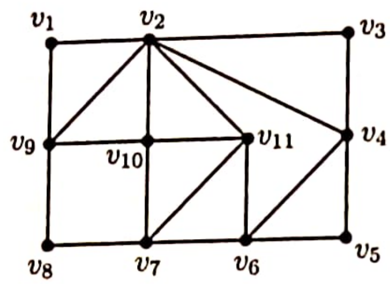


Figure 9.121

3. Show that the following graph does not contain an Euler circuit.

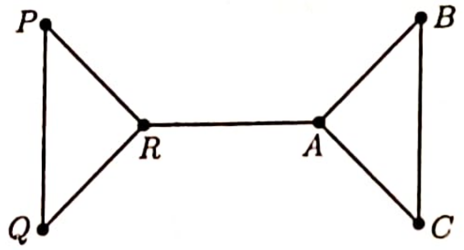


Figure 9.122

4. Show that the following graph contains an Euler trail.

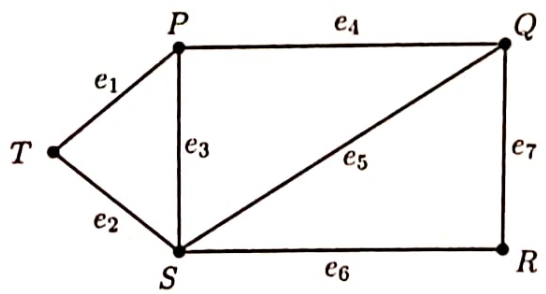


Figure 9.123

5. Prove that the complete bipartite graph $K_{2,3}$ contains an Euler trail.

6. Prove that the Petersen graph contains neither an Euler circuit nor an Euler trail.

7. Prove that a connected graph contains an Euler trail if and only if it has exactly zero or two vertices of odd degree.

Answers

1. The graph contains as an Euler circuit: *PAQBRQP*.

2. $v_1 v_2 v_9 v_{10} v_2 v_{11} v_7 v_{10} v_{11} v_6 v_4 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_1$

3. Starting with any vertex, it is not possible to return to that vertex without traversing the edge *RA* twice.

4. The graph contains $Pe_1Te_2Se_3Pe_4Qe_5Se_6Re_7Q$ as an Euler trail.

9.8.1 The Königsberg Bridge Problem

In the eighteenth century city named Königsberg in East Prussia (Europe), there flowed a river named Piegel River which divided the city into four parts. Two of these parts were the banks of the river and two were islands. These parts were connected with each other through seven bridges.

The citizens of the city seemed to have posed the following problem. By starting at any of the four land areas, can we return to that area after crossing *each* of the seven bridges *exactly once*?

This problem, now known as the *Königsberg Bridge problem*, remained unsolved for several years. In the year 1736, Euler analyzed the problem with the help of a graph and gave the solution. This was indeed the starting point for the development of graph theory.

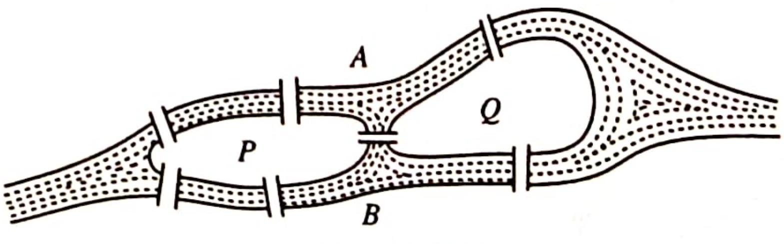


Figure 9.124

Let us see what the solution is (as given by Euler). Denote the land areas of the city by A, B, P, Q, where A, B are the banks of the river and P, Q are the islands (See Figure 9.124). Construct a graph by treating the four land areas as four vertices and the seven bridges connecting them as seven edges. The graph is as shown in Figure 9.125.

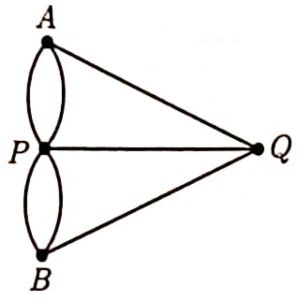


Figure 9.125

We note that, in this graph,

$$\deg(A) = \deg(B) = \deg(Q) = 3, \quad \deg(P) = 5$$

which are not even. Therefore, the graph does not have an Euler circuit*. This means that there does not exist a closed walk that contains all the edges exactly once. This amounts to saying that *it is not possible* to walk over each of the seven bridges exactly once and return to the starting point.

*See Theorem 1, Section 9.8.

Chapter 10

Trees

In this chapter we consider an important class of graphs, called *Trees*. Some basic properties of trees and their immediate consequences and related concepts are presented. The topic of *Prefix Codes* is introduced and illustrated.

10.1 Trees and their Basic properties

A graph G is said to be a ***tree*** *if it is connected and has no cycles.*

It immediately follows that a tree has to be a simple graph; because loops and parallel edges form cycles.

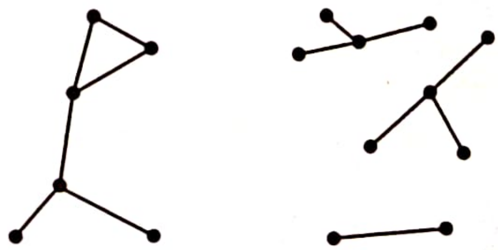
The graphs shown in Figure 10.1 are all trees. We observe that each of these trees possesses at least two pendant vertices. A pendant vertex of a tree is also called a ***leaf***.

(a) (b) (c)

Figure 10.1

A graph which is a tree is usually denoted by T (instead of G) to emphasize the structure.

The graphs shown in Figure 10.2 are not trees. Observe that the first of these contains a cycle whereas the second is not connected. However, each component of the second (disconnected) graph is a tree. Such a graph is called a ***forest***.



(a) (b)

Figure 10.2

The following theorems contain some basic properties of trees.

Theorem 1 . *In a tree, there is one and only one path between every pair of vertices.*

Proof: Let T be a tree. Then T is a connected simple graph. Since T is connected, there must be at least one path between every two vertices. If there are two paths between a pair of vertices of T, the union of the paths will become a cycle, and T cannot be a tree. Thus, between every pair of vertices in a tree there must exist one and only one path. •

Theorem 2 . *If in a graph G there is one and only one path between every pair of vertices, then G is a tree.*

Proof: Since there is a path between every pair of vertices in G, it is obvious that G is connected. Since there is only one path between every pair of vertices, G cannot have a cycle. Because, if there is a cycle, then there exist two paths between two vertices on the cycle. Thus, G is a connected graph containing no cycles. This means that G is a tree. •

The above two theorems may be combined together and put in the following form:

A graph G is a tree if and only if there is one and only one path between every pair of vertices in G.

Theorem 3 . *A tree with n vertices has n − 1 edges**.

Proof: We prove the theorem by induction on n.

The theorem is obvious for $n = 1$, $n = 2$ and $n = 3$; see the trees in Figure 10.1.

Assume that the theorem holds for all trees with n vertices where $n \leq k$, for a specified positive integer k.

Consider a tree T with $k + 1$ vertices. In T, let e be an edge with end vertices u and v. Since T is a tree, it has no cycles and therefore there exists no other edge or path between u and v. Hence, deletion of e from T will disconnect the graph and $T - e$ consists of exactly two components, say T_1 and T_2. Since T does not contain any cycle, the components T_1 and T_2 too do not contain any cycles. Hence, T_1 and T_2 are trees in their own right. Both of these trees have less than $k + 1$ vertices each, and therefore, according to the assumption made, the theorem holds for these trees; that is, each of T_1 and T_2 contains one less edge than the number of vertices in it. Therefore, since the total number of vertices in T_1 and T_2 (taken together) is $k + 1$, the total number of edges in T_1 and T_2 (taken together) is $(k + 1) - 2 = k - 1$. But T_1 and T_2 taken together is $T - e$. Thus, $T - e$ contains $k - 1$ edges. Consequently, T has exactly k edges.

Thus, if the theorem is true for a tree with $n \leq k$ vertices, it is true for a tree with $n = k + 1$ vertices. Hence, by induction, the theorem is true for all positive integers n. •

*In other words, for a tree of n vertices and m edges, we have $m = n - 1$. Said alternatively, this means that for a tree $T = (V, E)$, we have $|E| = |V| - 1$, or equivalently $|V| = |E| + 1$.

Theorem 4 . *Any connected graph with n vertices and n – 1 edges is a tree.*

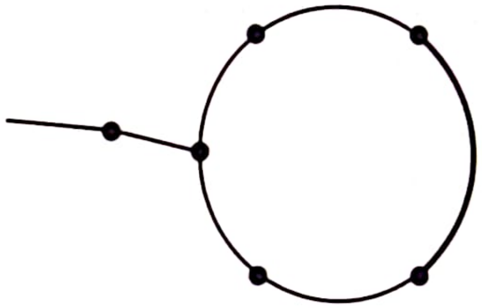


Figure 10.3

Proof: Let G be a connected graph with n vertices and $n-1$ edges. Assume that G is not a tree. Then G contains a cycle, say C. Let e be an edge in C. The graph G will not become disconnected if e is deleted (see Figure 10.3). Thus, $G-e$ is a connected graph. But, on the other hand, $G-e$ has n vertices and $n-2$ edges; therefore, it cannot be connected.* This is a contradiction. Hence, G must not have a cycle; this means that G must be a tree.

This completes the proof of the theorem. •

Remark: A disconnected graph with n vertices and $n-1$ edges need not be a tree; see the graph shown below (which is disconnected and has 4 vertices and 3 edges).

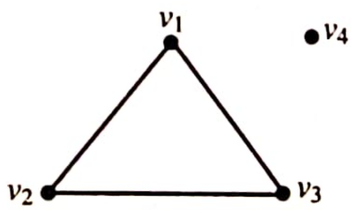


Figure 10.4

Theorems 3 and 4 can be put together in the following combined form:
A graph with n vertices is a tree if and only if it is connected and has n – 1 edges.

Theorem 5 . *A connected graph G is a tree if and only if adding an edge between any two vertices in G creates exactly one cycle in G.*

Proof: Suppose a connected graph G is a tree. Then G has no cycles and there is exactly one path between any two vertices, u, v. If we add an edge between u and v, then an additional path is created between u and v and the two paths constitute a cycle. Since G had no cycles earlier, this is the only cycle which G now possesses.

*Recall Theorem 2, Section 9.7.

Conversely, suppose G is connected and adding an edge between any two vertices u and v in G creates exactly one cycle in G. This implies that, before adding this edge, exactly one path was there between u and v. This implies that G is a tree.

The proof of the theorem is complete. •

Minimally connected graphs

A connected graph is said to be *minimally connected* if the removal of any one edge from it disconnects the graph.

For example, the graphs shown in Figure 10.1 are minimally connected. As already noted, all of these graphs are trees as well.

Theorem 6 . *A connected graph is a tree if and only if it is minimally connected.*

Proof: Suppose G is a connected graph which is not a tree. Then G contains a cycle C. The removal of any one edge e from this cycle will not make the graph disconnected. Therefore, G is not minimally connected. Thus, if a connected graph is not a tree then it is not minimally connected. This is equivalent to saying that if a connected graph is minimally connected then it is a tree (contrapositive).

Conversely, suppose G is a connected graph which is not minimally connected. Then there exists an edge e in G such that $G - e$ is connected. Therefore, e must be in some cycle in G. This implies that G is not a tree. Thus, if a connected graph is not minimally connected then it is not a tree. This is equivalent to saying that if a connected graph is a tree, then it is minimally connected (contrapositive).

This completes the proof of the theorem. •

Example 1 **(a)** *Show that the complete graph K_n is not a tree when $n > 2$.*

(b) *Show that the complete bipartite graph $K_{r,s}$ is not a tree when $r \geq 2$.*

► (a) If v_1, v_2, v_3 are any three vertices of K_n, where $n > 2$, then the closed walk $v_1v_2v_3v_1$ is a cycle in K_n. Since K_n has a cycle, it cannot be a tree.

(b) Let v_1 and v_2 be any two vertices in the first bipartite and v_1', v_2' be any two vertices in the other bipartite of $K_{r,s}$, with $s \geq r > 1$. Then, the closed walk $v_1v_1'v_2v_2'v_1$ is a cycle in $K_{r,s}$. Since $K_{r,s}$ has a cycle, it cannot be a tree. ■

Example 2 *Prove that a graph with n vertices, $n - 1$ edges and no cycles is connected.*

► Consider a graph G which has n vertices, $n - 1$ edges and no cycles. Suppose G is not connected. Let the components of G be H_i, $i = 1, 2, \ldots k$. If H_i has n_i vertices, we have $n_1 + n_2 + \cdots + n_k = n$. Since G has no cycles, H_is also do not have cycles. Further, they are all connected graphs. Therefore, they are trees. Consequently, each H_i must have $n_i - 1$ edges. Therefore, the total number of edges in these H_is is

$$(n_1 - 1) + (n_2 - 1) + \cdots + (n_k - 1) = n - k.$$

This must be equal to the total number of edges in G; that is $n - k = n - 1$. This is not possible, since $k > 1$. Therefore, G must be connected. ■

Example 3 *Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be two trees. If $|E_1| = 19$ and $|V_2| = 3|V_1|$, determine $|V_1|$, $|V_2|$, and $|E_2|$.*

► It is given that $|E_1| = 19$. Therefore,

$$|V_1| = |E_1| + 1 = 19 + 1 = 20.$$

Since $|V_2| = 3|V_1|$ as given, we get

$$|V_2| = 3|V_1| = 3 \times 20 = 60.$$

Lastly,

$$|E_2| = |V_2| - 1 = 60 - 1 = 59.$$ ■

Example 4 *If a tree has 2020 vertices, find the sum of the degrees of the vertices.*

► Since the number of vertices in the given tree is $|V| = 2020$, the number of its edges is $|E| = |V| - 1 = 2019$. Therefore (by hand-shaking property) the sum of degrees of its vertices $= 2|E| = 4038$. ■

Example 5 *Prove that a tree with two or more vertices contains at least two leaves (pendant vertices).*

► Consider a tree T with n vertices, where $n \geq 2$. Then it has $n - 1$ edges. Therefore (by the handshaking property), the sum of the degrees of the n vertices must be equal to $2(n-1)$. Thus, if $d_1, d_2, \ldots, d_n$ are the degrees of vertices of T, we have

$$d_1 + d_2 + \cdots + d_n = 2(n-1) = 2n - 2.$$

If each of $d_1, d_2, \ldots, d_n$ is ≥ 2, then their sum must be at least $2n$. Since this is not true, at least one of the d's is less than 2. Thus, there is a d which is equal to 1. (Since T is connected, no d can be zero). Without loss of generality, let us take this to be d_1. Then

$$d_2 + d_3 + \cdots + d_n = (2n-2) - 1 = 2n - 3.$$

This is possible only if at least one of $d_2, d_3, \ldots, d_n$ is equal to 1. So, there is at least one more d which is equal to 1. Thus, in T, there are at least two vertices with degree 1; that is, there are at least two pendant vertices (leaves). ■

Example 6 *Show that if a tree has exactly two pendant vertices, the degree of every non-pendant vertex is two.*

► Let n be the number of vertices in a tree T. Suppose it has exactly two pendant vertices (so that their degrees are 1 each). Let $d_1, d_2, \ldots, d_{n-2}$ be the degrees of the other (non-pendant) vertices. Then, since T has exactly $n-1$ edges, we have

$$1 + 1 + d_1 + d_2 + \cdots + d_{n-2} = 2(n-1)$$

$$\text{or} \qquad d_1 + d_2 + \cdots + d_{n-2} = 2n - 4 = 2(n-2)$$

The left hand side of this condition has $n-2$ terms, and none of these is one or zero. Therefore, this condition holds only if each of the d_is is equal to two. ■

Example 7 *Show that, in a tree, if the degree of every non-pendant vertex is 3, the number of vertices in the tree is an even number.*

► Let n be the number of vertices in a tree T. Of these, let k be the number of pendant vertices. Then, if each non-pendant vertex is of degree 3, the sum of the degrees of vertices is $k+3(n-k)$. This must be equal to $2(n-1)$, by the handshaking property. Thus,

$$k + 3(n-k) = 2(n-1), \quad \text{or} \quad n = 2(k-1).$$

This shows that n is an even number. ■

Example 8 *Suppose that a tree T has two vertices of degree 2, four vertices of degree 3 and three vertices of degree 4. Find the number of pendant vertices in T.*

► Let N be the number of pendant vertices in T. It is given that T has two vertices of degree 2, four vertices of degree 3 and three vertices of degree 4. Therefore,

Total number of vertices $= N + 2 + 4 + 3 = N + 9$, and

Sum of the degrees of vertices $= (N \times 1) + (2 \times 2) + (4 \times 3) + (3 \times 4) = N + 28$.

Since T has $N+9$ vertices, it has $N + 9 - 1 = N + 8$ edges.

Therefore, by handshaking property, we have $N + 28 = 2(N+8)$ which gives $N = 12$.

Thus, the given tree has 12 pendant vertices. ■

Example 9 *If a tree T has four vertices of degree 2, one vertex of degree 3, two vertices of degree 4 and one vertex of degree 5, find the number of leaves in T.*

► Let N be the number of leaves (pendant vertices) in T. Then:

$$\text{Total No. of vertices} = N + 4 + 1 + 2 + 1 = N + 8$$

$$\text{Sum of the degrees of vertices} = (N \times 1) + (4 \times 2) + (1 \times 3) + (2 \times 4) + (1 \times 5) = N + 24.$$

Since T has $N+8$ vertices, it has $N+7$ edges. Therefore, by hand shanking property.

$$N + 24 = 2(N+7) \quad \text{which gives} \quad N = 10.$$

Thus, the given tree has 10 leaves. ■

Example 10 *Suppose that a tree T has N_1 vertices of degree 1, N_2 vertices of degree 2, N_3 vertices of degree $3, \ldots, N_k$ vertices of degree k. Prove that*

$$N_1 = 2 + N_3 + 2N_4 + 3N_5 + \cdots + (k-2)N_k.$$

► From what is given, we note that, in T,

Total number of vertices $= N_1 + N_2 + \cdots + N_k$, and

Sum of the degrees of vertices $= N_1 + 2N_2 + 3N_3 + 4N_4 + 5N_5 + \cdots + kN_k$.

Therefore, the total number of edges in T is $N_1 + N_2 + \cdots + N_k - 1$, and the handshaking property gives

$$N_1 + 2N_2 + 3N_3 + 4N_4 + 5N_5 + \cdots + kN_k = 2(N_1 + N_2 + \cdots + N_k - 1)$$

Rearranging terms, this gives

$$N_3 + 2N_4 + 3N_5 + \cdots + (k-2)N_k = N_1 - 2.$$

This is the required result. ■

Exercises

1. Prove that a graph with n vertices is a tree if and only if it has $n-1$ edges and no cycles.
2. Show that a tree with exactly two leaves must be a path.
3. If a tree has four vertices of degree 3, two vertices of degree 4 and one vertex of degree 5, show that it should have 10 pendant vertices.
4. In a tree with 14 pendant vertices, the degree of every non-pendant vertex is either 4 or 5. Show that the tree has 3 vertices of degree 4 and 2 vertices of degree 5.
5. In a tree with $r + s$ vertices, if r vertices are pendant vertices and s vertices have degree 4 each, prove that $2s = r - 2$.
6. Prove that, in a tree, the minimum number of pendant vertices is equal to the maximum degree of a vertex.
7. Prove that every tree with two or more vertices is 2-chromatic.
8. Prove that in a tree with $n (\geq 2)$ vertices the number of distinct paths is ${}^nC_2$.
9. Let T be a tree with n vertices, where $n \geq 4$, and v be a vertex of maximum degree in T. Prove that T is a path if and only if $d(v) = 2$.
10. Let T be a tree with n vertices, $n \geq 3$. Show that there is a vertex v of T with degree at least two such that every vertex adjacent to v, except possibly one, is a pendant vertex.

10.2 Rooted Trees

Let D be a *directed graph* and G be its underlying graph. We say that D is a ***directed tree*** whenever G is a tree. Thus, a directed tree is a directed graph whose underlying graph is a tree.

A directed tree T is called a ***rooted tree*** if (i) T contains a unique vertex, called the *root*, whose in-degree is equal to 0, and (ii) the in-degrees of all other vertices of T are equal to 1.

Figures 10.5 and 10.6 depict two directed trees. The first of these is not a rooted tree whereas the second is a rooted tree.

Figure 10.5

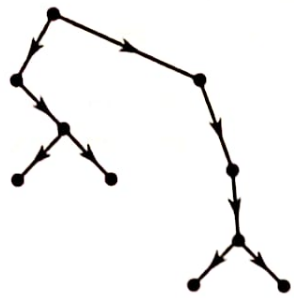


Figure 10.6

In a rooted tree, we denote the root by r and draw the (diagram of the) tree downward from an upper level to a lower level, so that the arrows can be dropped. Then the root r will be at the uppermost level (zeroth level) and all other vertices will be at lower levels.

A vertex v (other than the root r) of a rooted tree is said to be at the *k-th level* or has ***level number k*** if the path from r to v is of length k. If v_1 and v_2 are two vertices such that v_1 has a lower level number than v_2 and there is a path from v_1 to v_2, then we say that v_1 is an ***ancestor*** of v_2, or that v_2 is a ***descendant*** of v_1. In particular, if v_1 and v_2 are such that v_1 has a lower level number than v_2 and there is an edge (- directed edge, actually) from v_1 to v_2, then v_1 is called the ***parent*** of v_2, or v_2 is called the ***child*** of v_1. Two vertices with a common parent are referred to as ***siblings***.

In a rooted tree a vertex whose out-degree is 0 is called a ***leaf*** and a vertex which is not a leaf is called an ***internal vertex***.

For example, suppose we redraw the directed tree of Figure 10.6 as shown below without arrows (- which are understood) and with vertices labeled.

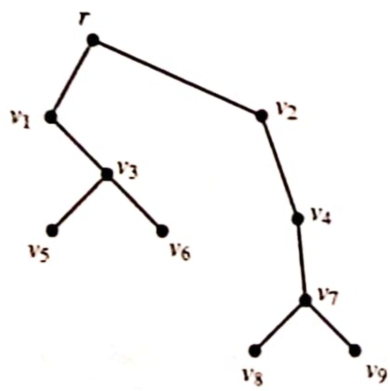


Figure 10.7

In this rooted tree, we note that

(1) v_1 and v_2 are at the *first level*, v_3, v_4 are at the *second level*, v_5, v_6, v_7 are at the *third level*, and v_8 and v_9 are at the *fourth level*.

(2) v_1 is the *ancestor* of v_3, v_5, v_6 (or v_3, v_5, v_6 are the *descendants* of v_1), and v_2 is the *ancestor* of v_4, v_7, v_8, v_9 (or v_4, v_7, v_8, v_9 are the *descendants* of v_2).

(3) v_1 is the *parent* of v_3 (or v_3 is a *child* of v_1).

(4) v_5 and v_6 are *siblings*, and v_8 and v_9 are *siblings*.

(5) v_5, v_6, v_8, v_9 are *leaves*, and all other vertices are *internal vertices*.

m-ary Tree

A rooted tree T is called an *m-ary tree* if every internal vertex of T is of out-degree $\leq m$; that is if every internal vertex of T has at most m children.

A rooted tree T is called a ***complete m-ary tree*** if every internal vertex of T is of out-degree m; that is if every internal vertex of T has exactly m children.

Binary Tree

An m-ary tree for which $m = 2$ is called a *binary tree*.

In other words, a rooted tree T is called a *binary tree* if every vertex of T is of out-degree ≤ 2; that is if every vertex has at most two children.

A complete m-ary tree for which $m = 2$ is called a ***complete binary tree***.

In other words, a rooted tree T is called a complete binary tree if every internal vertex of T is of out-degree 2; that is if every internal vertex has exactly two children.

The rooted tree shown in Figure 10.7 is a binary tree; it is not a complete binary tree. The rooted tree shown in Figure 10.8 is a complete binary tree.

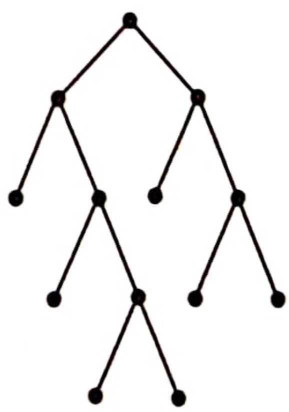


Figure 10.8

Balanced Tree

If T is a rooted tree and h is the largest level number achieved by a leaf of T, then T is said to have *height* h. A rooted tree of height h is said to be *balanced* if the level number of every leaf is h or $h-1$.

The tree shown in Figure 10.7 is of height 4 and is balanced too. The tree shown in Figure 10.8 is of height 4 but is not balanced.

Full binary Tree

Let T be a complete binary tree of height h. Then T is called a ***full binary tree*** if all the leaves in T are at level h.

The complete binary tree shown in Figure 10.8 is not a full binary tree. The tree shown in Figure 10.9 is a full binary tree.

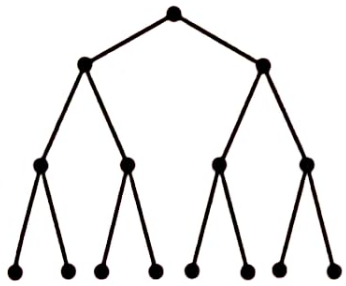


Figure 10.9

Example 1 *Let T be a complete m-ary tree of order n with p leaves and q internal vertices. Prove the following:*

(a) $n = mq + 1 = \dfrac{mp-1}{m-1}$

(b) $p = (m-1)q + 1 = \dfrac{(m-1)n+1}{m}$

(c) $q = \dfrac{n-1}{m} = \dfrac{p-1}{m-1}$

► We recall that a complete m-ary tree T is a rooted tree in which every internal vertex is of out-degree m.

Since every vertex in a rooted tree is either a leaf or an internal vertex, the total number of vertices in T (that is, the order of T) is the sum of the number of leaves in T and the number of internal vertices in T. Thus, we have

$$n = p + q \tag{i}$$

Since the out-degree of every leaf of T is zero and the out-degree of every internal vertex of T is m,

The sum of the out-degrees of vertices of T

$$= (p \times 0) + (q \times m) = qm \tag{ii}$$

Since in a rooted tree of order n, the in-degree of the root is zero and the in-degrees of the remaining $(n-1)$ vertices are 1 each,

The sum of the in-degrees of vertices of T

$$= (1 \times 0) + ((n-1) \times 1) = n - 1 \qquad \text{(iii)}$$

By the First Theorem of Digraph Theory*, the two sums given by (ii) and (iii) must be equal. Thus, $qm = n - 1$, or

$$n = qm + 1, \qquad \text{(iv)}$$

or $$q = \frac{n-1}{m} \qquad \text{(v)}$$

Using expression (iv) in expression (i), we get $qm + 1 = p + q$, which gives

$$p = (m-1)q + 1 \qquad \text{(vi)}$$

This result readily yields

$$q = \frac{p-1}{m-1} \qquad \text{(vii)}$$

Putting this into expression (iv) we get

$$n = m\left(\frac{p-1}{m-1}\right) + 1 = \frac{m(p-1) + (m-1)}{m-1} = \frac{mp-1}{m-1} \qquad \text{(viii)}$$

Lastly, putting expression (vii) into expression (i), we get

$$n = p + \frac{p-1}{m-1}, \quad \text{or} \quad (m-1)n = p(m-1) + p - 1 = pm - 1$$

or $$p = \frac{(m-1)n+1}{m} \qquad \text{(ix)}$$

Thus, all the required results are proved. ■

Remark: In the case of a *complete binary tree* (for which $m = 2$), the results of the above Example become

(a) $n = 2q + 1 = 2p - 1$

(b) $p = q + 1 = \frac{1}{2}(n+1)$

(c) $q = \frac{1}{2}(n-1) = p - 1$

---

*See Section 9.1.

Example 2 *Find the number of vertices and the number of leaves in a complete binary tree having* 10 *internal vertices.*

► Let n be the number of vertices in the tree being considered. Since this tree has $q = 10$ internal vertices, the expression $n = 2q + 1$ yields $n = 21$. Consequently, the number of leaves is $p = n - q = 11$. ■

Example 3 (a) *Find the number of internal vertices in a complete 5-ary tree with* 817 *leaves.*

(b) *Find the number of leaves in a complete 6-ary tree of order* 733.

► (a) From Example 1, we have

$$q = \frac{p-1}{m-1}$$

For $m = 5$ and $p = 817$, this gives $q = 204$. Thus, the given tree has 204 internal vertices.

(b) From Example 1, we have

$$p = \frac{(m-1)n+1}{m}$$

For $m = 6$ and $n = 733$, this gives $p = 611$. Thus, the given tree has 611 leaves. ■

Example 4 *The computer laboratory of a school has* 10 *computers that are to be connected to a wall socket that has* 2 *outlets. Connections are made by using extension cords that have* 2 *outlets each. Find the least number of cords needed to get these computers set up for use.*

► Here, the wall socket may be regarded as the root of a complete binary tree having the computers as its leaves and the internal vertices, other than the root, as extension cords.

Then the number of leaves in the tree is $p = 10$. Therefore, the number of internal vertices in the tree is $q = p - 1 = 10 - 1 = 9$. Accordingly, the number of extension cords needed (namely the number of internal vertices minus the root) is $q - 1 = 8$. (See Figure 10.10 for illustration).

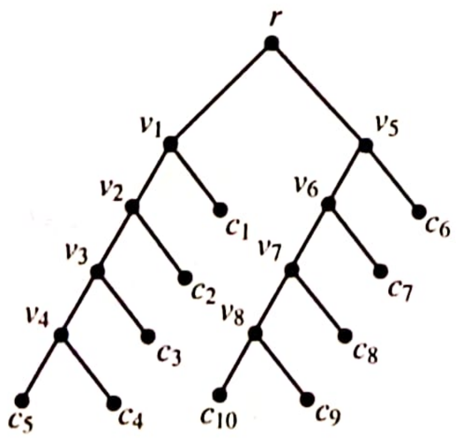


Figure 10.10 ■

Example 5 *A class room contains 25 microcomputers that must be connected to a wall socket that has 4 outlets. Connections are made by using extension cords that have 4 outlets each. Find the least number of cords needed to get this computer set up for the class.*

▶ Here, the wall socket may be regarded as the root of a complete 4-ary tree with the computers as its leaves and the internal vertices other than the root as extension cords.

Thus, here, $m = 4$ and $p = 25$. Therefore, the number of internal vertices is (see Example 1)

$$q = \frac{p-1}{m-1} = \frac{24}{3} = 8.$$

Consequently, the number of extension cords needed (namely the number of internal vertices minus the root) is $q - 1 = 7$. ■

Exercises

1. Which of the following is a rooted tree?

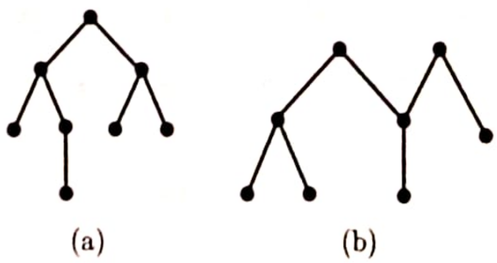


Figure 10.11

2. For the rooted tree shown below, find the levels of the vertices A, H, F, M.

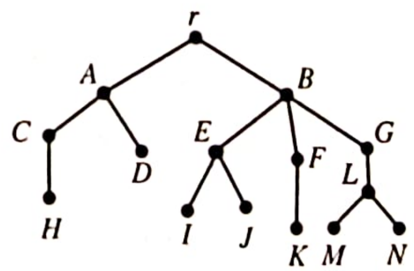


Figure 10.12

3. In the rooted tree shown in Figure 10.12, identify

   (i) all ancestors of L, (ii) all descendants of A,

   (iii) the children of B, (iv) the parent and siblings of C.

4. In the tree shown in Figure 10.13, identify the following:

(i) vertices which are leaves (ii) the root (iii) the parent of g
(iv) descendants of c (v) siblings of s (vi) vertices having the level number 4.

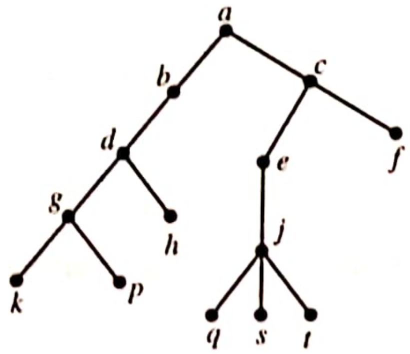


Figure 10.13

5. Which of the following is a binary tree? complete binary tree?

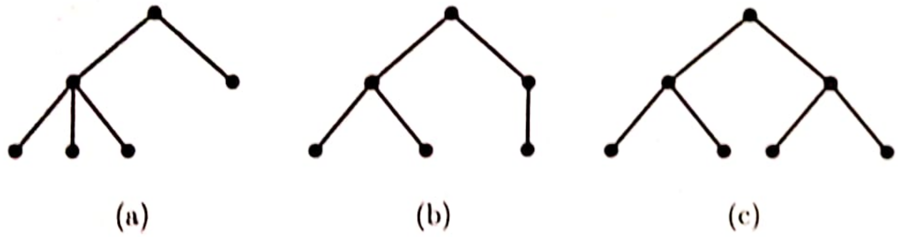


Figure 10.14

6. Which of the following is a balanced tree? full binary tree?

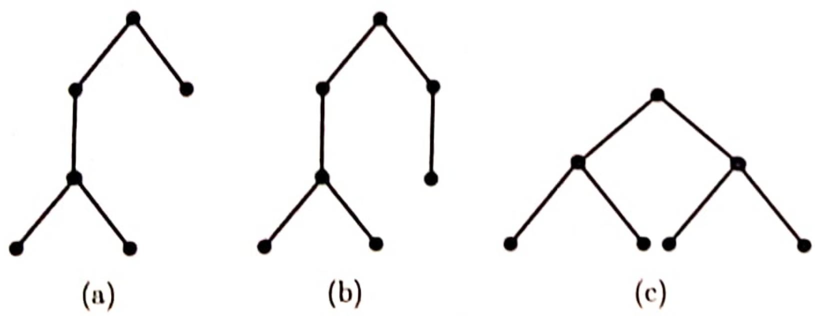


Figure 10.15

7. A complete binary tree has 20 leaves. How many vertices does it have?

8. Find the number of leaves in a complete binary tree if it has 29 vertices.

9. In a single-elimination singles tennis tournament, a player is eliminated after a single loss. If 32 players compete in the tournament, how many matches must be played to determine the number-

one player? (Hint : Consider a complete binary tree with the number-one player as the root, matches as internal vertices and the competitors as leaves).

10. A complete 3-ary tree T has 34 internal vertices. How many leaves does T have?

Answers

1. (a) rooted tree, (b) not a rooted tree. **2.** $A:1$, $H:3$, $F:2$, $M:4$.

3. (i) r, B, G (ii) C, D, H (iii) E, F, G (iv) parent:A, sibling:D

4. (i) k, p, h, q, s, t, f (ii) a (iii) d (iv) e, f, j, q, s, t (v) q, t (vi) k, p, q, s, t

5. (a) not a binary tree (b) binary tree, but not complete (c) complete binary tree.

6. (a) not a balanced tree (b) balanced tree, but not full (c) full binary tree.

7. 39 **8.** 15 **9.** 31 **10.** 69

10.2.1 Sorting

Suppose we wish to sort (– rearrange/reorganize) a given list of n integers in nondecreasing order. The most common (and the easiest) way of carrying out this sorting consists of two parts. In the first part, we recursively split the given list and all subsequent lists in half (or as close as possible to half) until each sublist contains a single element. In the second part, we merge the sublists in nondecreasing order until the original n integers have been sorted. The splitting and merging process is done by the use of balanced complete trees. This method of sorting a list is known as ***Merge Sort***.

Example 1 *Using the merge-sort method, sort the list* 7, 3, 8, 4, 5, 10, 6, 2, 9.

► First, we recursively split the given list and all subsequent lists in half or as close as possible to half until each sublist contains a single element. This splitting process is represented in the tree shown in Figure 10.16(a).

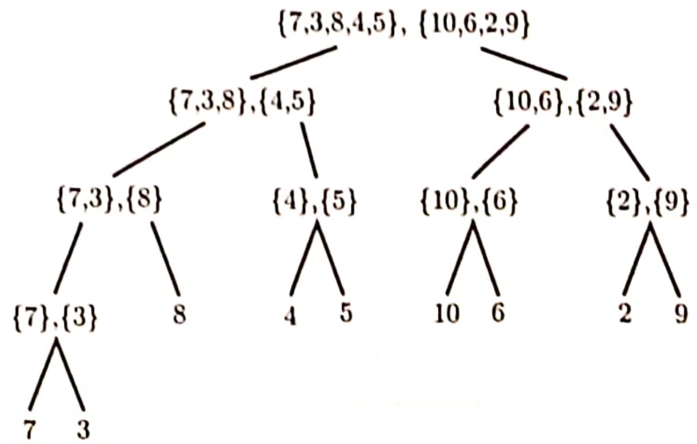


Figure 10.16: (a)

Now we merge the sublists in nondecreasing order until the items in the original list have been sorted. This merging process is represented in Figure 10.16(b).

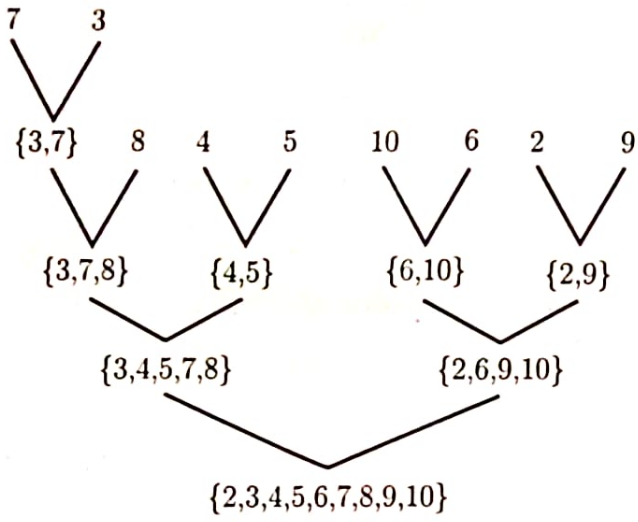


Figure 10.16: (b)

Thus, the sorted form of the given list is 2, 3, 4, 5, 6, 7, 8, 9, 10. ■

Example 2 *Apply merge-sort to the list* $-1, 7, 4, 11, 5, -8, 15, -3, -2, 6, 10, 3$.

▶ The given list is first split into sublists as depicted in the tree shown in Figure 10.17(a).

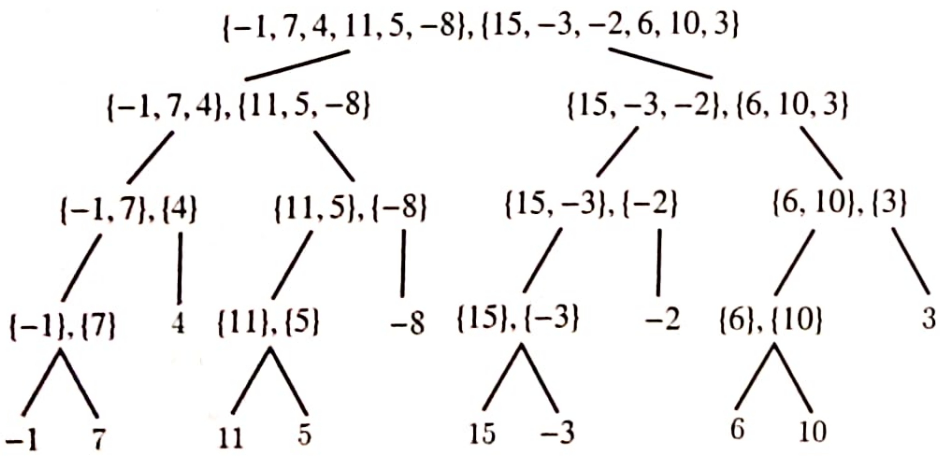


Figure 10.17: (a)

The sublists obtained above are now merged as depicted in Figure 10.17(b).

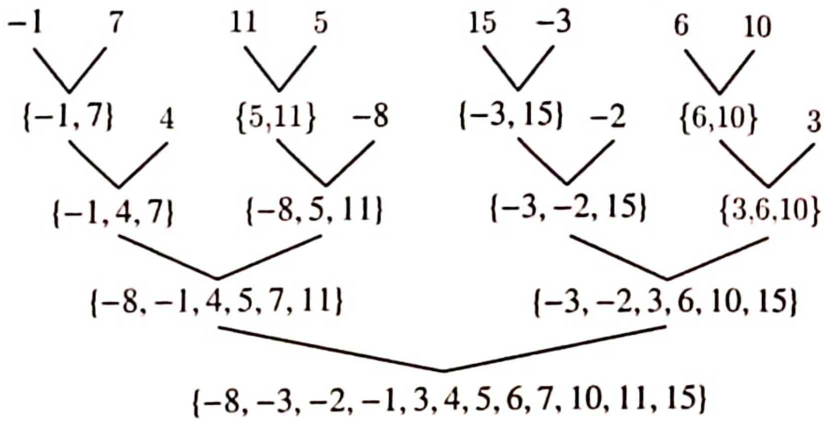


Figure 10.17: (b)

Thus, the sorted-out version of the given list is −8, −3, −2, −1, 3, 4, 5, 6, 7, 10, 11, 15. ■

Exercises

Apply the merge-sort to each of the following lists:

1. 6, 2, 7, 3, 4, 9, 5, 1, 8.

2. −1, 0, 2, −2, 3, 6, −3, 5, 1, 4.

3. −9, 6, 5, −3, 4, 2, −7, 6, −5, 10, −11, 0, 1.

10.3 Prefix codes and Weighted trees

Recall that a sequence is a set whose elements are listed (arranged) in order as the first element, second element, third element, and so on. The number of elements contained in a sequence is called its *length*.

A sequence consisting of only 0 and 1 is called a ***binary sequence*** or a ***binary string***.

Thus,

$$01, 001, 101, 11001, 1000100$$

are all binary sequences with lengths 2, 3, 3, 5, 7, respectively.

Binary sequences are used as codes for messages sent through transmitting channels.

Suppose a message consisting of the letters a, e, n, r, t is to be transmitted. Suppose we use the following coding scheme for these letters:

$$a : 1, \quad e : 0, \quad n : 10, \quad r : 01, \quad t = 101.$$

Under this coding scheme, suppose the message *eat* is to be transmitted. Then, the coded form of the message is given by the binary sequence 01101. Now, the question is: can this binary sequence be decoded correctly (without ambiguity)? The answer is: No!. Because the sequence 01101 can be decoded not only as *eat* but also as *rt*, or *rar*, or *eaar* (for example). Accordingly, the coding scheme indicated above is of no use in transmissions.

Alternatively, suppose we use the following coding scheme:

$$a:10, \quad e:0, \quad n:1101, \quad r:111, \quad t:1100.$$

Then, the message *eat* is to be transmitted as the binary sequence 0101100. We can check that this sequence, when decoded, yields only *eat* and no other message.

We observe that in the first of the coding schemes considered above, the code 1 is assigned to a, the code 10 is assigned to n and the code 101 is assigned to t and that the code assigned to t contains the codes assigned to a and n as prefixes. This is why there arises ambiguity in decoding. In the second of the coding schemes, the code of any letter is *not a prefix* of the code of any other letter. This is why there cannot be ambiguity in decoding. Coding schemes of this type are of interest in coding theory and a brief account of the same is given in the following paragraphs.

Prefix Codes

Let P be a set of binary sequences that represent a set of symbols. Then P is called a *prefix code*† if no sequence in P is the prefix of any other sequence in P.

For example, the sets

$$P_1 = \{10, 0, 1101, 111, 1100\}, \quad \text{and} \quad P_2 = \{000, 001, 01, 10, 11\}$$

are prefix codes, whereas the sets

$$A_1 = \{01, 0, 101, 10, 1\} \quad \text{and} \quad A_2 = \{1, 00, 01, 000, 0001\}$$

are *not* prefix codes.

Prefix codes can be represented by binary trees as illustrated below.

Consider the prefix code

$$P_2 = \{000, 001, 01, 10, 11\}$$

indicated above. In this code, the longest sequence has length 3. Keeping this in mind, let us construct a *full binary tree* of height 3, and assign the symbol 0 to every edge that is directed towards the child in the left from its parent vertex and 1 to every edge that is directed towards the child in the right, as shown in Figure 10.18*.

†prefix codes are also called *Huffman codes*, in honor of their inventor D. A. Huffman.

*Assigning 0 and 1 to edges of a tree according to the scheme indicated and illustrated here is conventional in nature and is called *labeling*. The tree got after labeling is called the *labeled tree*.

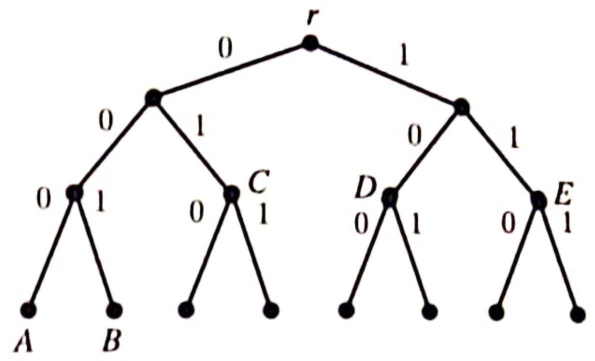


Figure 10.18

The five sequences present in P_2 can now be identified with five vertices of the above tree. We note that the vertex marked A can be reached from the root r through the three edges labeled 0, 0, 0. Accordingly, the vertex A can be assigned the sequence 000. Similarly, the vertex marked B can be assigned the sequence 001, the vertex marked C can be assigned the sequence 01, the vertex marked D can be assigned the sequence 10, and the vertex marked E can be assigned the sequence 11. Thus, all the five sequences present in P_2 can be assigned to the five vertices, marked A, B, C, D, E, of the tree being considered.

The subtree extracted from the full binary tree of Figure 10.18 that contains the root r and the vertices A, B, C, D, E is shown in Figure 10.19. This subtree represents the prefix code given by the set P_2.

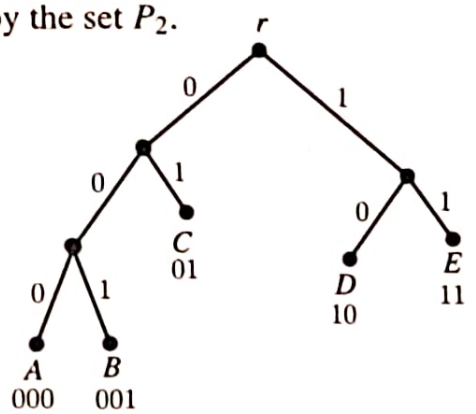


Figure 10.19

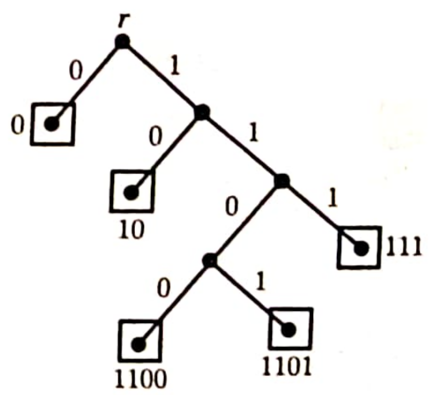


Figure 10.20

The binary tree representing the prefix code given by the set P_1 (indicated earlier) is shown in Figure 10.20. The vertices of this tree to which the sequences in P_1 can be assigned are boxed.

Example 1 *Consider the prefix code:*

$$a : 111, \quad b : 0, \quad c : 1100, \quad d : 1101, \quad e : 10.$$

Using this code, decode the following sequences:

(i) 1001111101, (ii) 10111100110001101, (iii) 1101111110010.

► Keeping the given code in mind, we first split the given sequences into appropriate number of parts. Then, the decoded version of the sequence is written down.

(i) Splitting: 10 0 111 1101, Decode: *e b a d*

(ii) Splitting: 10 111 10 0 1100 0 1101, Decode: *e a e b c b d*

(iii) Splitting: 1101 111 1100 10, Decode: *d a c e* ∎

Example 2 *A code for* $\{a, b, c, d, e\}$ *is given by*

$$a : 00, \quad b : 01, \quad c : 101, \quad d : x10, \quad e : yz1,$$

where $x, y, z \in \{0, 1\}$. *Determine* x, y *and* z *so that the given code is a prefix code.*

► If we set $x = 0$, then the code for d reads 010 which contains the code 01 for b as a prefix. Therefore, we cannot take $x = 0$. This implies that $x = 1$. Then, $d : 110$.

If we set $y = 0$, $z = 0$ then the code for e reads 001, which contains the code for a as a prefix. If we set $y = 0$, $z = 1$, then the code for e reads 011, which contains the code for b as a prefix. If we set $y = 1$, $z = 0$, then the code for e becomes identical with that for c. Hence, we take $y = 1$, $z = 1$ so that $e : 111$.

With the choices of x, y, z as obtained above the given code reads

$$a : 00, \quad b : 01, \quad c : 101, \quad d : 110, \quad e : 111.$$

This is evidently a prefix code. ∎

Example 3 *Construct the binary tree that represents the prefix code obtained in Example 2.*

► The prefix code considered in Example 2 is

$$P = \{00, 01, 101, 110, 111\}.$$

The longest possible sequence present in this code is of length 3. As such, the required binary tree is of height 3. We draw this tree in such a way that its leaves can be identified by the sequences in P through the labeling rule. The tree is shown in the Figure 10.21.

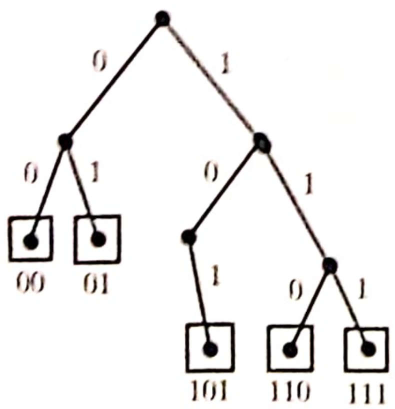


Figure 10.21 ∎

Example 4 *Obtain the prefix code represented by the following labeled complete binary tree:*

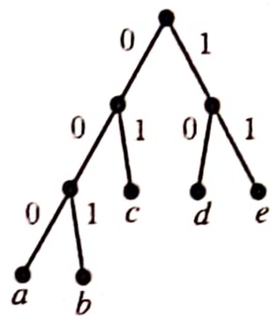


Figure 10.22

► The leaves of the given tree are represented by the symbols a, b, c, d, e. These leaves are identified by the sequences as indicated in the following Table:

Leaf:	a	b	c	d	e
Sequence:	000	001	01	10	11

This table determines the required prefix code as

$$P = \{000, 001, 01, 10, 11\}$$ ■

Example 5 *Find the prefix codes for the letters* B, E, I, K, L, T, P, S *if the coding scheme is as shown in Figure* 10.23.

Hence

(a) *Find the codes for the words* PIPE *and* BEST

(b) *Decode the strings*

(i) 000011100001 **(ii)** 1111111101101011110

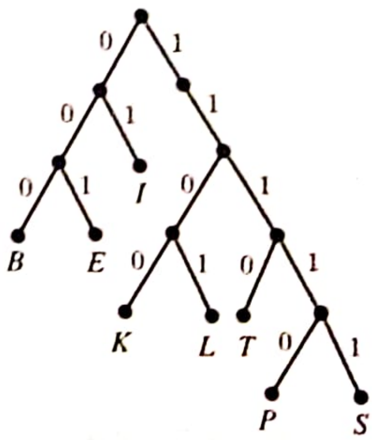


Figure 10.23

► The given letters are the leaves of the given tree. The codes for these leaves are indicated below.

Leaf :	B	E	I	K	L	T	P	S
Code :	000	001	01	1100	1101	1110	11110	11111

(a) It follows that the code for the PIPE is

$$111100111110001$$

and the code for the word BEST is

$$000001111111110.$$

(b) Keeping the codes obtained above in mind, we first split the given strings as shown below:

(i) 000 01 1100 001 **(ii)** 11111 11110 1101 01 1110

It now follows that the decoded versions of the these strings are BIKE and SPLIT respectively. ■

Weighted Trees

Consider a set of n positive integers $w_1, w_2, \ldots, w_n$, where $w_1 \le w_2 \le \ldots \le w_n$. Suppose we assign these integers to the n leaves of a complete binary tree $T = (V, E)$ in any one-to-one manner. The resulting tree is called a *complete, weighted, binary tree* with $w_1, w_2, \ldots w_n$ as *weights*. If $l(w_i)$ is the level number of the leaf of T to which the weight w_i is assigned, then $W(T)$ defined by

$$W(T) = \sum_{i=1}^{n} w_i \, l(w_i)$$

is called the ***weight*** of the tree T.

Figure 10.24(a) shows a complete binary tree T_1 to whose leaves the weights 3, 5, 7, 8 are assigned, and Figure 10.24(b) shows another complete binary tree T_2 to whose leaves the same weights are assigned. We find that

$$W(T_1) = (8 \times 3) + (7 \times 3) + (5 \times 2) + (3 \times 1) = 58,$$

and

$$W(T_2) = (8 \times 2) + (7 \times 2) + (3 \times 2) + (5 \times 2) = 46.$$

This illustrates the fact that for a given set of weights, the value of $W(T)$ depends upon the tree T chosen.

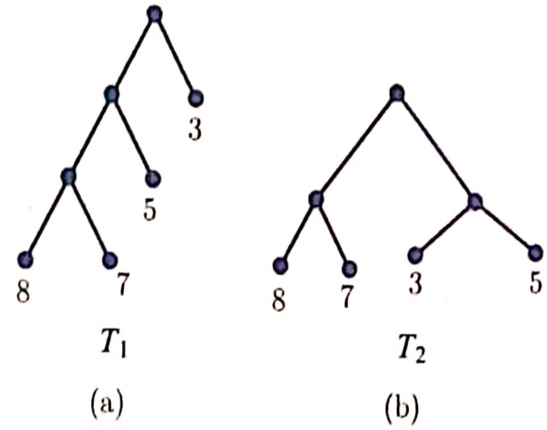


Figure 10.24

Optimal Tree

Given a set of weights, suppose we consider the set of all complete binary trees to whose leaves these weights are assigned. A tree in this set which carries the minimum weight is called an ***optimal tree*** for the weights. For a given set of weights, there can be more than one optimal tree.

The construction of an optimal tree for a given set of weights is illustrated below.

Let $\{4, 15, 25, 5, 8, 16\}$ be a set of six weights. Let us first arrange these six weights in non-decreasing order and assign them to six isolated vertices A, B, C, D, E, F as shown in Figure 10.25(i). The weights are indicated in brackets.

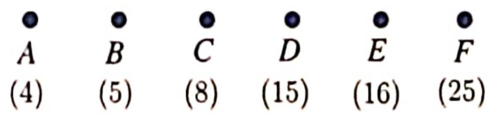


Figure 10.25: (i)

We note that the vertices A and B carry the smallest weights, 4 and 5. Add these weights to get the weight 9 and assign it to a new vertex v_1. Draw a tree having v_1 as the root and A and B as its children. Rearrange the vertices present at this stage (namely the new vertex v_1 and the old vertices C, D, E, F) in the non-decreasing order of their weights. The resulting graph is shown in Figure 10.25(ii).

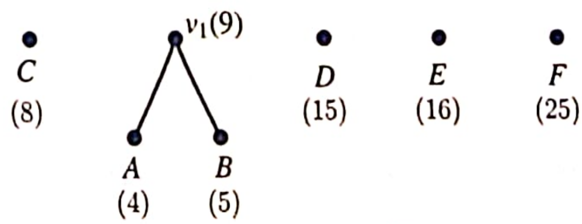


Figure 10.25: (ii)

Now, among the vertices C, v_1, D, E, F, two vertices which carry the smallest weights are C and v_1. Add their weights (8 and 9) to get a new weight 17. Assign this weight to a new vertex v_2 and draw a tree having v_2 as the root and C and v_1 as its children. Rearrange the vertices present at this stage (namely v_2, D, E, F) in the non-decreasing order of their weights. The resulting graph is shown below.

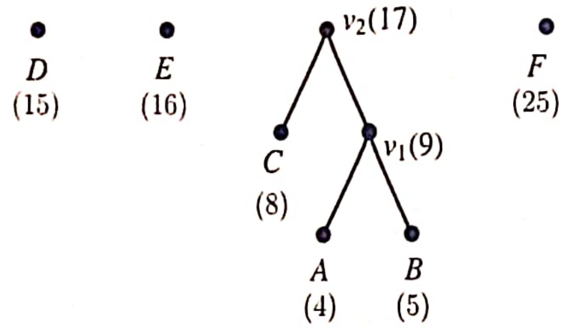


Figure 10.25: (iii)

Among the vertices D, E, v_2, F, the two vertices which carry minimum weights are D and E. Add their weights (15 and 16) to get a new weight 31. Assign this weight to a new vertex v_3 and draw a tree having v_3 as the root and D and E as its children. Rearrange the vertices present at this stage (namely v_2, v_3 and F) in the non-decreasing order of their weights. The resulting graph is shown below.

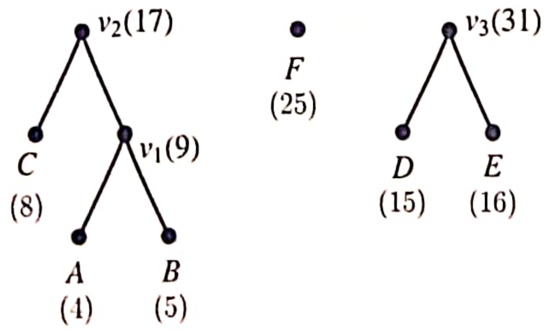


Figure 10.25: (iv)

Among the vertices v_2, F, v_3, the two vertices which carry smallest weights are v_2 and F. Add their weights (17 and 25) and get a new weight 42. Assign this weight to a new vertex v_4 and draw a tree having v_4 as the root and v_2 and F as its children. Rearrange the vertices v_3 and v_4 in the non-decreasing order of their weights. The resulting graph is shown below.

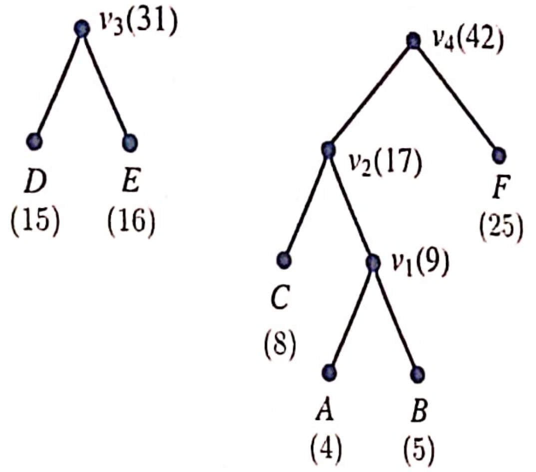


Figure 10.25: (v)

The disconnected graph shown above consists of two trees having v_3 and v_4 as roots. Add their weights (31 and 42) and get a new weight 73. Assign this weight to a new vertex r and draw a tree having r as the root and v_3 and v_4 as its children. This tree is shown in Figure 10.26.

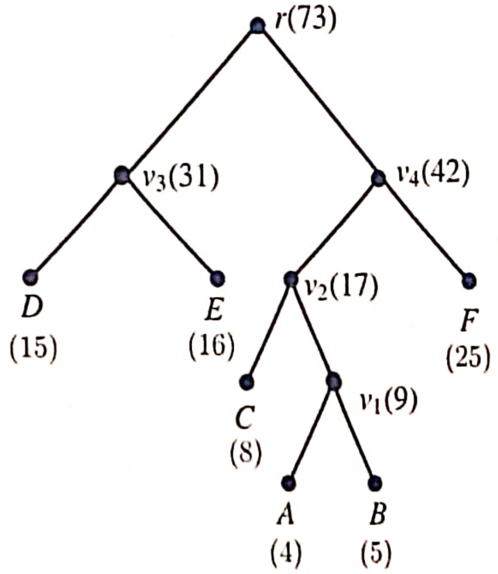


Figure 10.26

The tree obtained as explained above (that is the tree shown in Figure 10.26) is a complete, weighted binary tree whose leaves are the vertices A, B, C, D, E, F with which we started. This tree serves as an optimal tree for the weights considered.

Keeping the weights of A, B, C, D, E, F in mind and noting their levels in the above tree, we find that the weight of the optimal tree is

$$W(T) = (4 \times 4) + (5 \times 4) + (8 \times 3) + (15 \times 2) + (16 \times 2) + (25 \times 2) = 172.$$

The procedure adopted in constructing an optimal tree as illustrated above is known as *Huffman's procedure*. The tree itself is often called an *Huffman's tree*. This tree is *not unique.**

Optimal Prefix Code

Huffman tree (optimal tree) can be used to obtain a prefix code for the symbols representing its leaves. For this purpose, we first label symbols 0 and 1 to its edges by the labeling procedure indicated earlier. Then all the vertices and the leaves of the tree can be identified by binary sequences. The binary sequences through which the leaves are identified yield a prefix code for the symbols representing these leaves. This prefix code is known as an ***optimal prefix code.***

For example, let us consider the Huffman tree of Figure 10.26 and assign the symbols 0 and 1 to its edges according to the usual labeling rule. The resulting labeled tree is shown below:

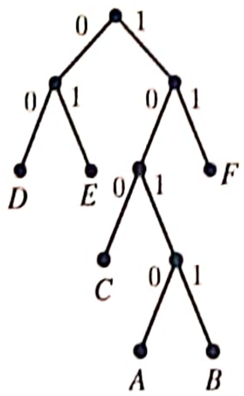


Figure 10.27

The leaves of the above tree and the binary sequences that identify these leaves are shown in the following Table:

Leaf:	A	B	C	D	E	F
Binary sequence:	1010	1011	100	00	01	11

This Table displays an optimal prefix code for the symbols A, B, C, D, E, F. Since the optimal tree is not unique, the optimal prefix code is also not unique.

Example 6 *Construct an optimal prefix code for the symbols a, o, q, u, y, z that occur with frequencies* 20, 28, 4, 17, 12, 7, *respectively.*

▶ Treating the given frequencies as the weights and the corresponding symbols as the isolated vertices, we first arrange the symbols such that their frequencies are in nondecreasing order. This is shown below:

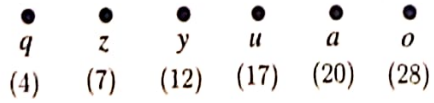


Figure 10.28: (i)

*This is because of multiple choices one may have in the rearrangement of vertices at different steps and the choice in selecting left and right subtrees.

Now, we construct an optimal tree having q, z, y, u, a, o as leaves by using the Huffman's procedure. The construction is done step-by-step and the graphs obtained in these steps are shown below in the order of their occurrence. (See Figures 10.28(ii)-(vi)). The final graph is labeled (using the labeling procedure) and depicted in Figure 10.29.

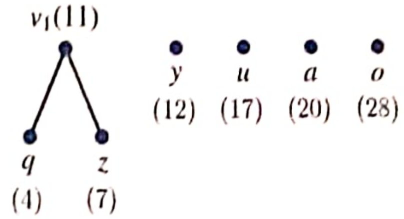


Figure 10.28: (ii)

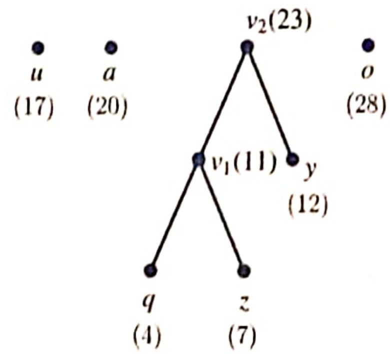


Figure 10.28: (iii)

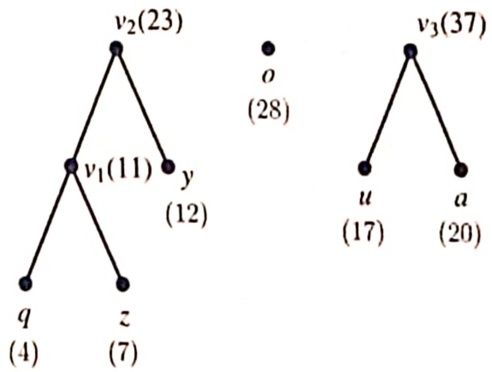


Figure 10.28: (iv)

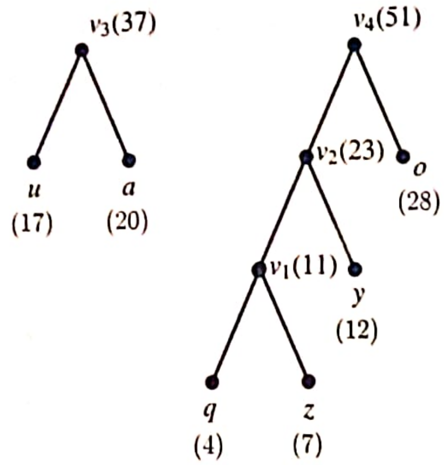


Figure 10.28: (v)

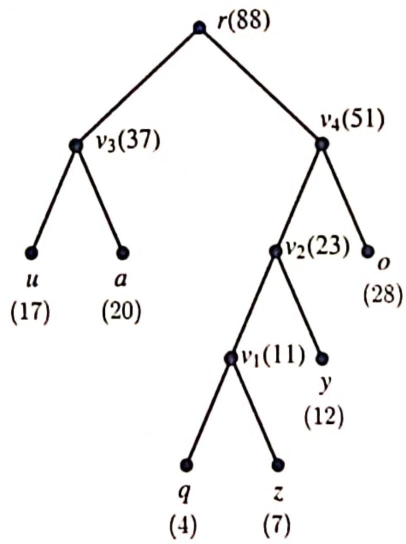


Figure 10.28: (vi)

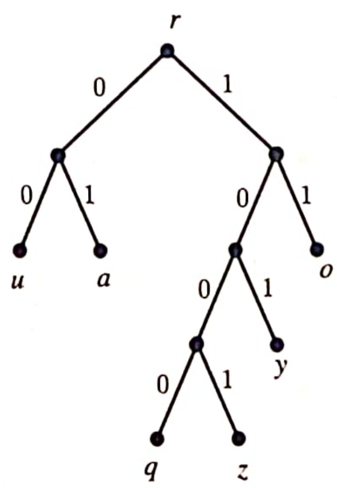


Figure 10.29

The tree shown in Figure 10.29 is an Huffmann tree (optimal tree) for the given data. From this graph, we obtain the following optimal prefix code for the given symbols:

$$a : 01, \quad o : 11, \quad q : 1000, \quad u : 00, \quad y : 101, \quad z : 1001.$$ ■

Example 7 *Construct an optimal prefix code for the symbols A, B, C, D, E, F, G, H, I, J that occur with respective frequencies* 78, 16, 30, 35, 125, 31, 20, 50, 80, 3.

▶ First, we arrange the given symbols in the non-decreasing order of their weights (frequencies). Their representation as isolated vertices is shown below:

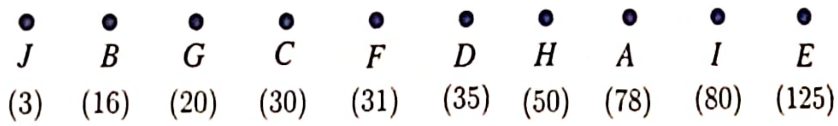


Figure 10.30: (i)

We now construct an optimal tree having $A, B, C, D, E, F, G, H, I, J$ as leaves by using the Huffman procedure. The construction is done step-by-step and the graphs obtained in these steps are shown below in Figures 10.30(ii)-(x) in the order of their occurrence. The final graph is labeled and depicted in Figure 10.31.

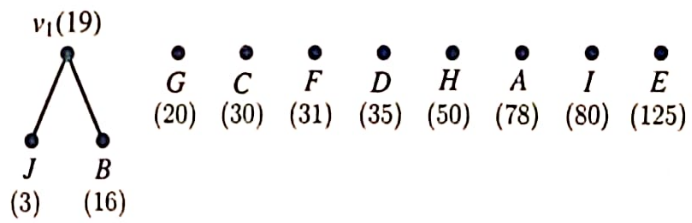


Figure 10.30: (ii)

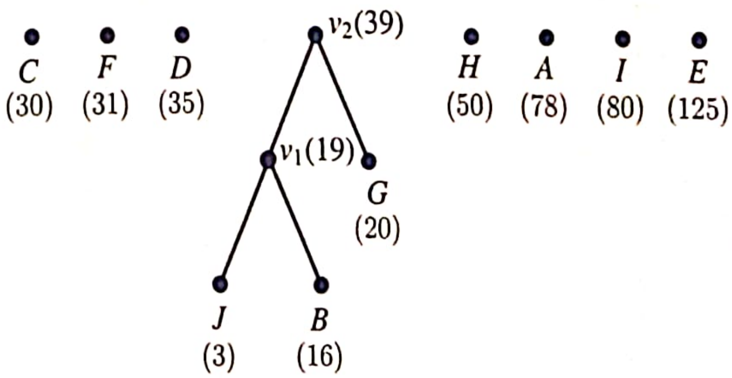


Figure 10.30: (iii)

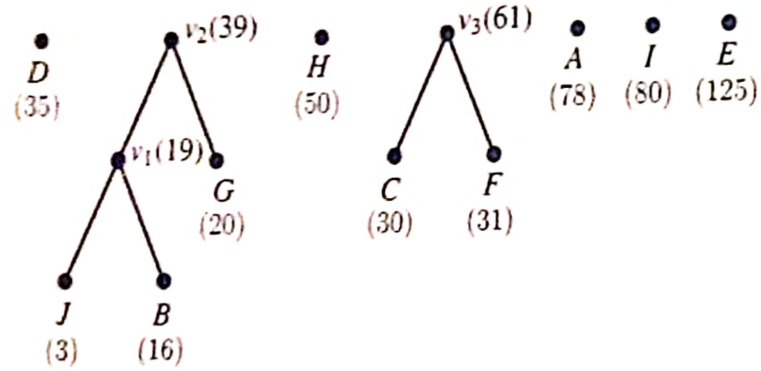


Figure 10.30: (iv)

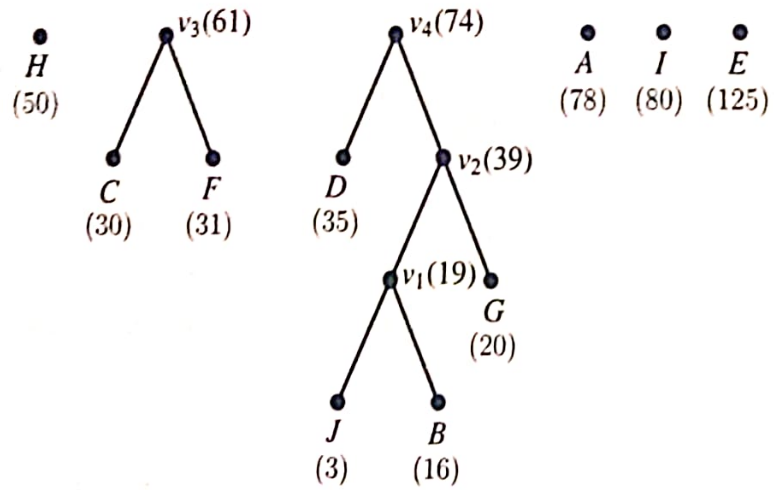


Figure 10.30: (v)

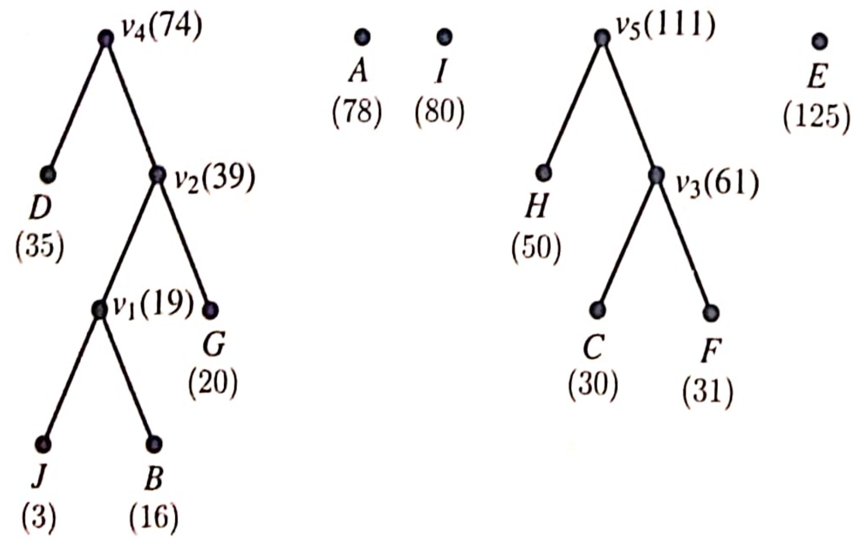


Figure 10.30: (vi)

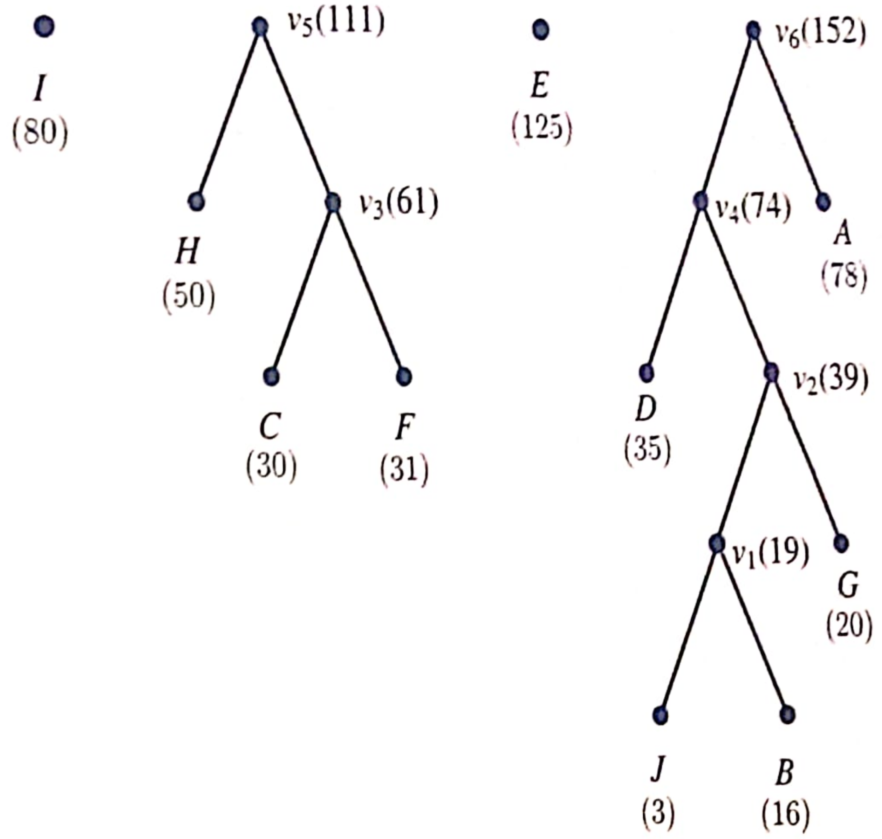


Figure 10.30: (vii)

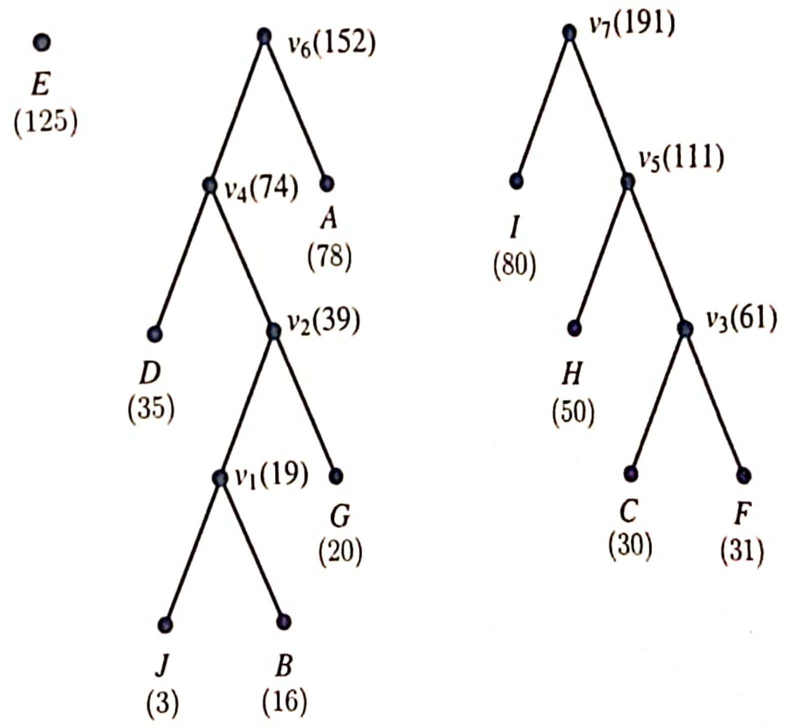


Figure 10.30: (viii)

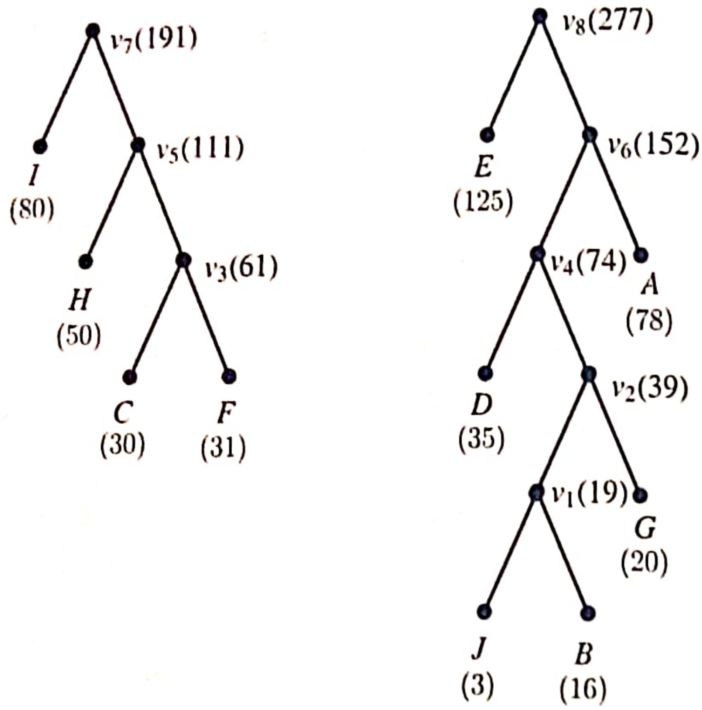


Figure 10.30: (ix)

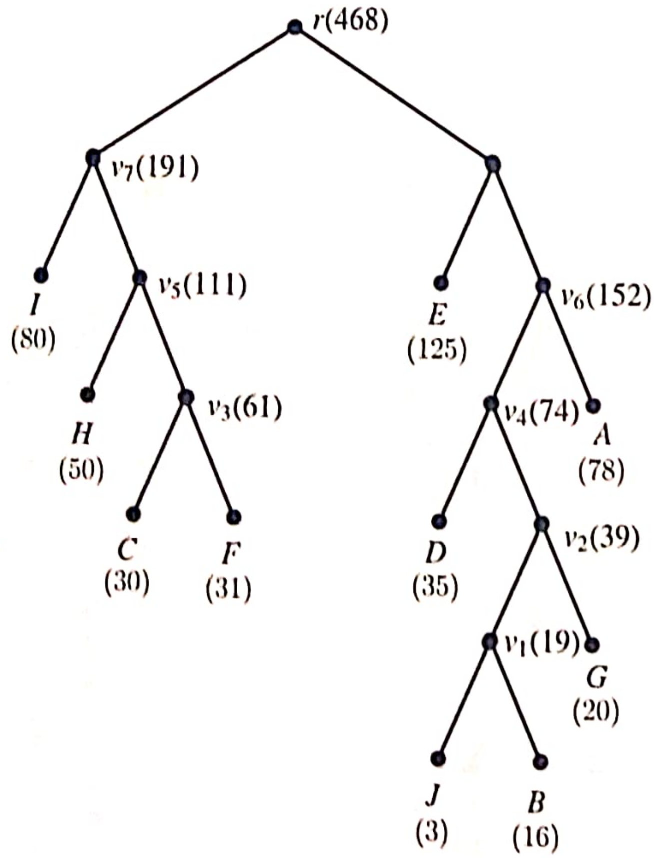


Figure 10.30: (x)

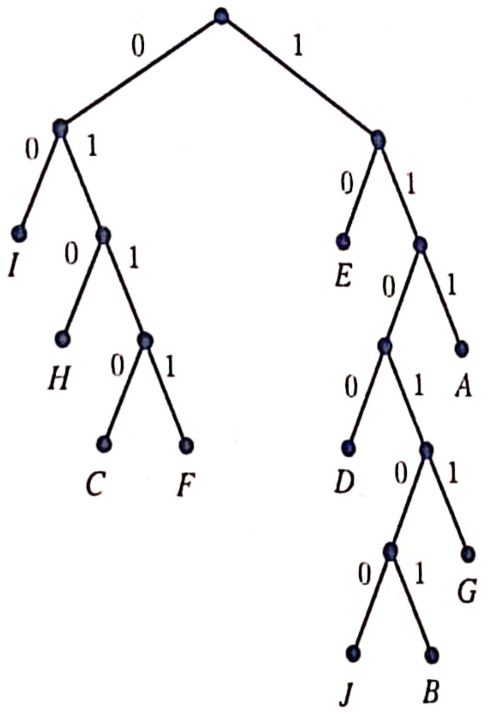


Figure 10.31

The tree in Figure 10.31 is an Huffmann tree (optimal tree) for the given data. From this tree, we obtain the following optimal prefix code for the given symbols.

A: 111, B: 110101, C: 0110, D: 1100, E: 10
F: 0111, G: 11011, H: 010, I: 00, J: 110100. ∎

Example 8 *Construct an optimal prefix code for the letters of the word* ENGINEERING. *Hence deduce the code for this word.*

▶ The given word consists of the letters E, N, G, I, R, with frequencies 3, 3, 2, 2, 1 respectively. First, we arrange these letters in the non-decreasing order of their frequencies (which may be regarded as weights). Their representation as isolated verties is shown below.

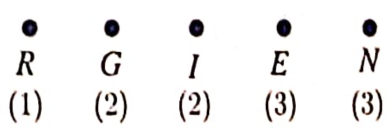


Figure 10.32: (i)

We now construct an optimal tree having these letters as leaves by using the Huffman's procedure. The graphs obtained in successive steps of the procedure are shown below in Figures 10.32(ii)-(v) in the order of their occurrence. The labeled version of the final tree is shown in Figure 10.33.

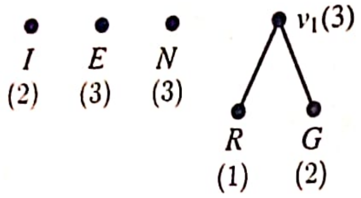


Figure 10.32: (ii)

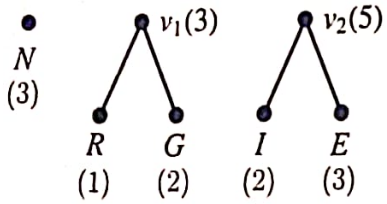


Figure 10.32: (iii)

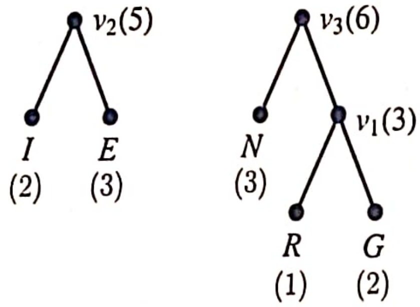


Figure 10.32: (iv)

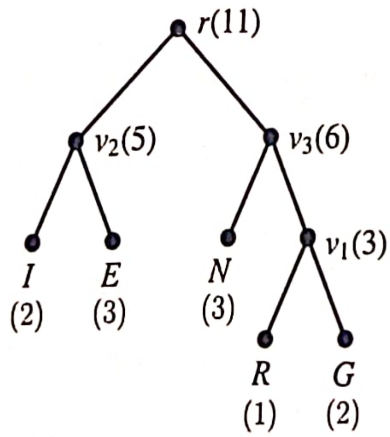


Figure 10.32: (v)

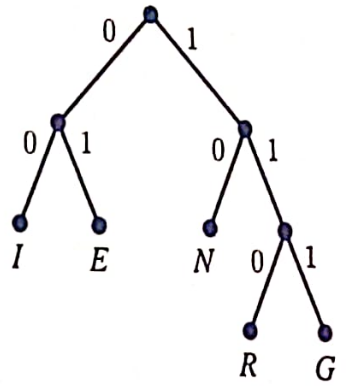


Figure 10.33

The tree shown in Figure 10.33, is the optimal tree that we sought. From this tree, we obtain the optimal prefix codes shown below for the letters with which we started:

$$R:110 \quad G:111 \quad I:00 \quad E:01 \quad N:10$$

Accordingly, the code for the given word ENGINEERING is

$$0110111001001011100010111$$ ■

Example 9 *Obtain an optimal prefix code for the message* LETTER RECEIVED. *Indicate the code.*

▶ The given message consists of letters L, E, T, R, C, I, V, D with frequencies 1, 5, 2, 2, 1, 1, 1, 1 respectively. Further, there is one blank space (□) between the two words of the message.

First, we arrange the letters and □ in the non-decreasing order of their weights (frequencies). Their representation as isolated vertices is shown below

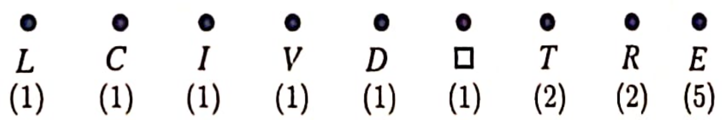


Figure 10.34: (i)

We now construct an optimal tree having these symbols as leaves by using the Huffman's procedure. The graphs obtained in successive steps of the procedure are shown below in Figures 10.34(ii)-(ix) in the order of their occurrence. The labeled version of the final graph is shown in Figure 10.35.

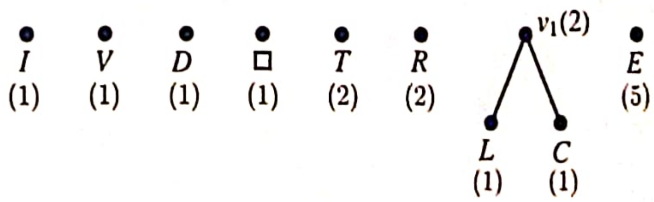


Figure 10.34: (ii)

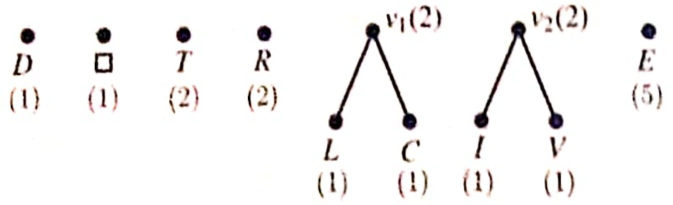


Figure 10.34: (iii)

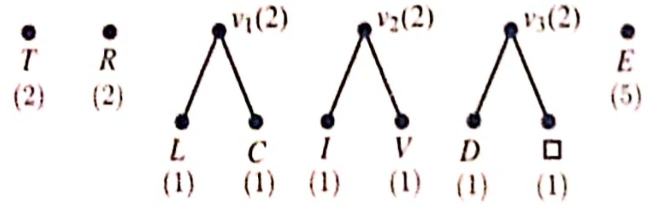


Figure 10.34: (iv)

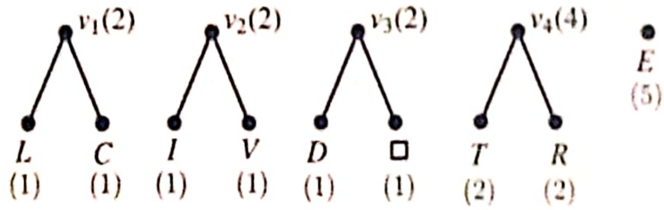


Figure 10.34: (v)

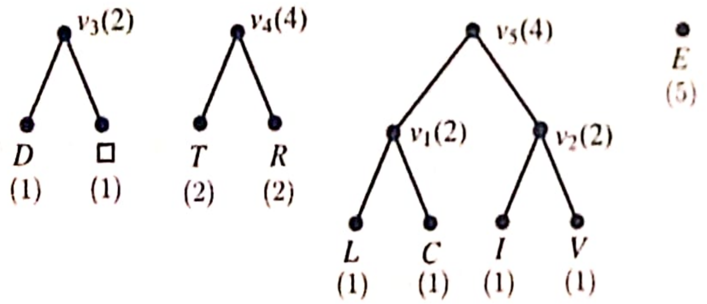


Figure 10.34: (vi)

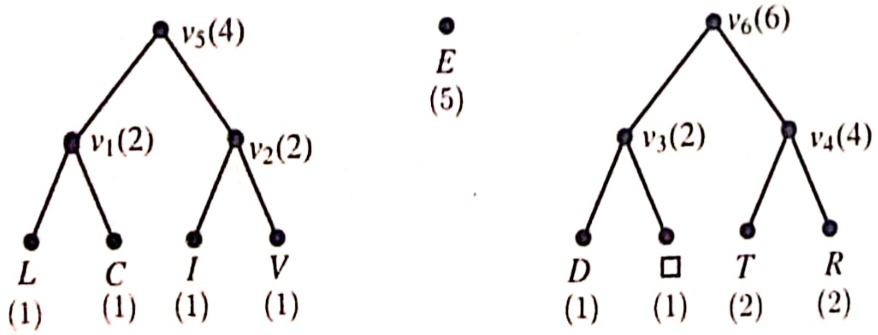


Figure 10.34: (vii)

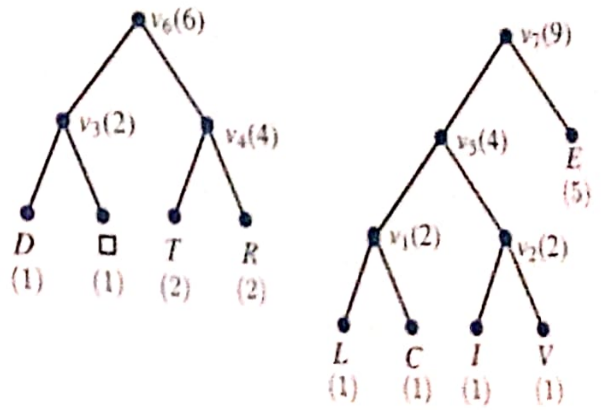


Figure 10.34: (viii)

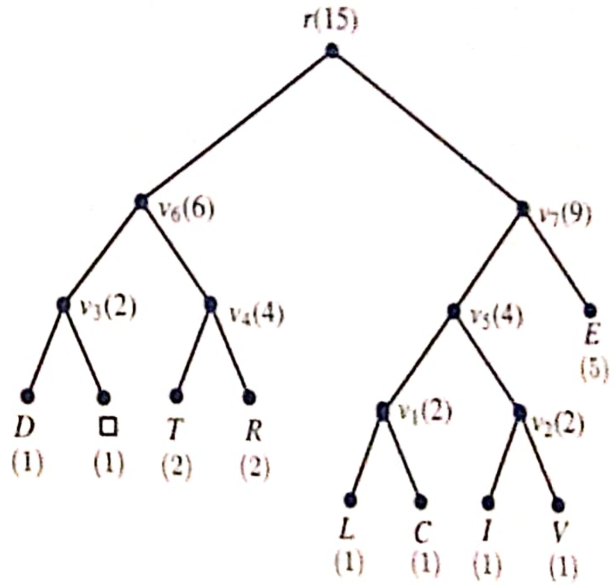


Figure 10.34: (ix)

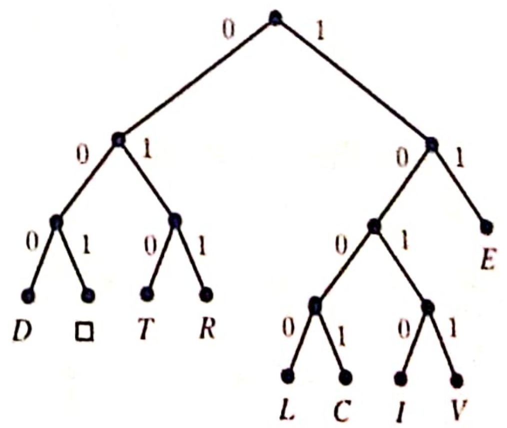


Figure 10.35

The tree in Figure 10.35 is the optimal tree that we sought. From this tree we obtain the following optimal prefix codes for the symbols with which we started.

$$L:1000 \quad C:1001 \quad I:1010 \quad V:1011$$

$$D:000 \quad \square:001 \quad T:010 \quad R:011 \quad E:11$$

Accordingly, the code for the given message LETTER RECEIVED is:

$$10001101001011011001011111001111010101111000$$ ■

Example 10 *Obtain an optimal prefix code for the message* ROAD IS GOOD. *Indicate the code.*

➤ The given message consists of letters R, O, A, D, I, S, G with frequencies 1, 3, 1, 2, 1, 1, 1, respectively. Further, there is a blank space (□) occuring twice (– one □ between ROAD and IS, and another □ between IS and GOOD).

First, we arrange the letters and □ in the non-decreasing order of their weights (frequencies). Their representation as isolated vertices is shown below:

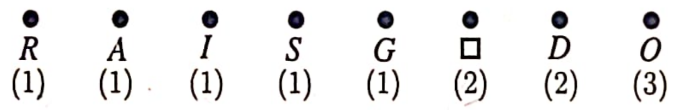


Figure 10.36: (i)

We now construct an optimal tree having these symbols as leaves by using the Huffman's procedure. The graphs obtained in successive steps of the procedure are shown below in Figures 10.36(ii)-(viii) in the order of their occurrence. The labelled version of the final tree is shown in Figure 10.37.

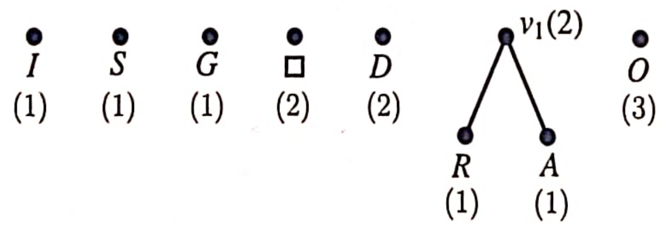


Figure 10.36: (ii)

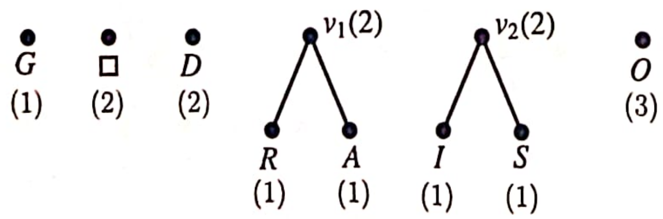


Figure 10.36: (iii)

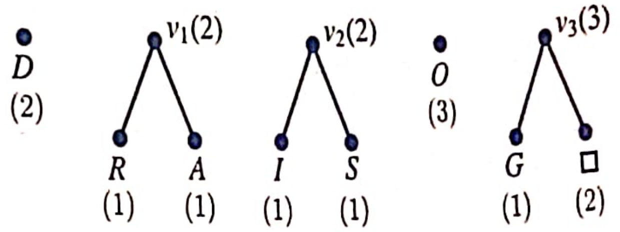


Figure 10.36: (iv)

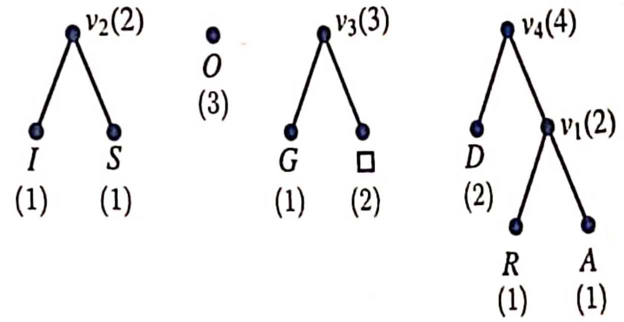


Figure 10.36: (v)

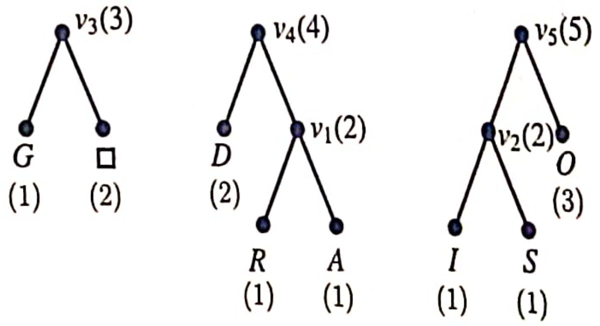


Figure 10.36: (vi)

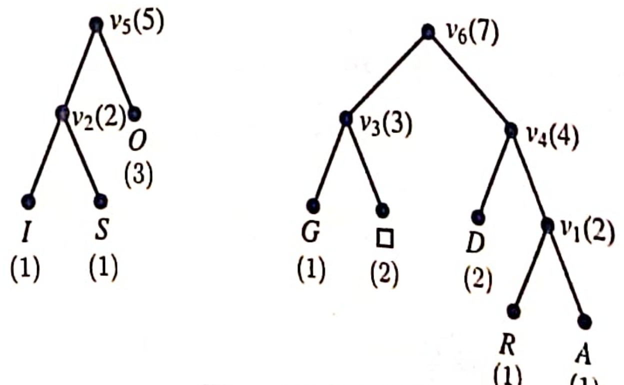


Figure 10.36: (vii)

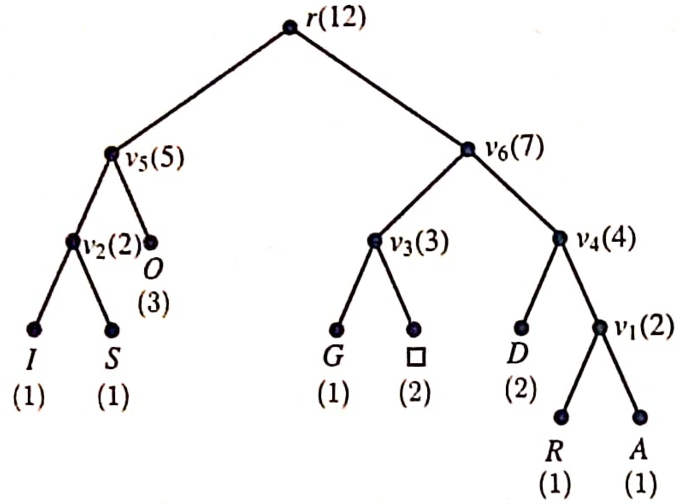


Figure 10.36: (viii)

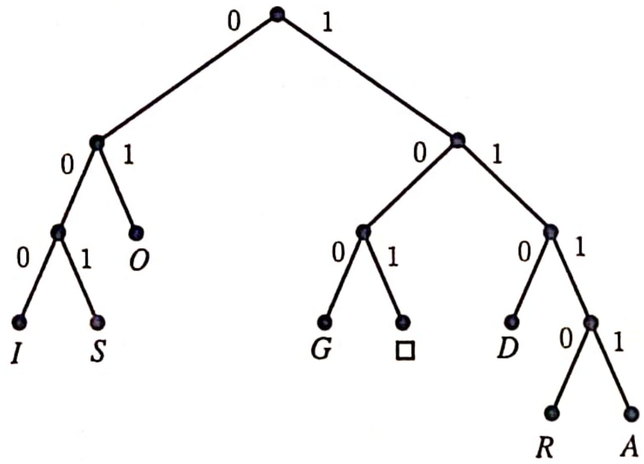


Figure 10.37

The tree shown in Figure 10.37 is the optimal tree that we sought. From this tree, we obtain the following optional prefix code for the symbols with which we started.

R: 1110, A: 1111, I: 000, S: 001,
G: 100, □ : 101, D: 110 O: 01

Accordingly, the code for the given message ROAD IS GOOD is

11100111111010100000110110001011110 ∎

Exercises

1. Draw the binary tree that represents the prefix code considered in Worked Example 1.

2. Which of the following sets represent prefix codes? Represent them by binary trees.

$$C = \{00,\ 010,\ 011,\ 10,\ 111,\ 1100,\ 1101\}$$

$$A = \{00,\ 011,\ 111,\ 11011\}$$

$$B = \{1100,\ 1010,\ 1111,\ 101\}$$

$$D = \{10,\ 111,\ 1100,\ 11010,\ 11011\}$$

3. Find the prefix code represented by the following tree:

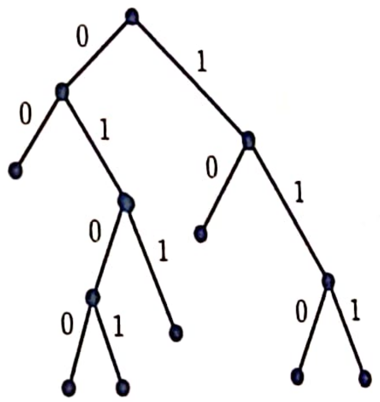


Figure 10.38

4. Find the weights of the complete binary trees T_1, T_2, T_3 shown below:

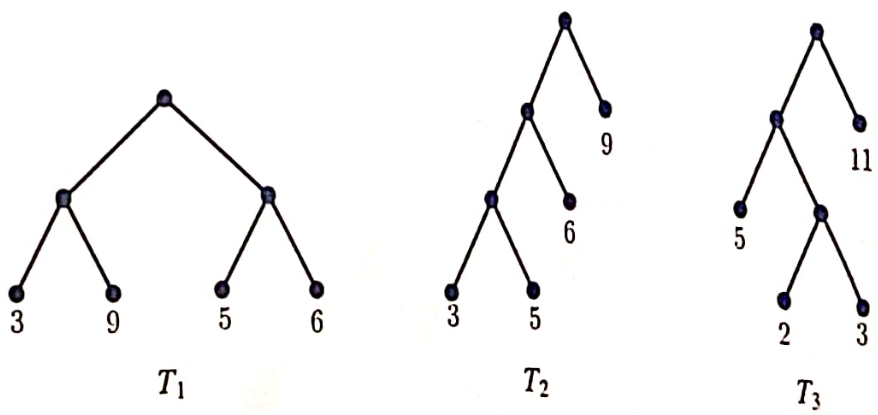


Figure 10.39

5. Construct an optimal tree for the weights 2, 3, 5, 10, 10.

6. Construct an optimal tree for the weights 11, 9, 12, 14, 10, 24 and 16.

7. Construct an optimal prefix code for the symbols with given weights in each of the following cases:

(1)

symbol:	a	b	c	d	e	f	g
weight:	10	30	5	15	20	15	5

(2)

symbol:	A	B	C	D	E	F	G	H
weight:	22	5	11	19	2	11	25	5

8. Obtain an optimal prefix code for the letters of the word CALCULUS. Indicate the code for the word.

9. Obtain an optimal prefix code for the message TAKE CARE. Indicate the code of the message.

10. Obtain an optimal prefix code for the message PROPOSAL ACCEPTED. Indicate the code for the message.

11. Obtain an optimal prefix code for the message MISSION SUCCESSFUL. Indicate the code for the message.

12. Obtain an optimal prefix code for the message FALL OF THE WALL. Indicate the code for the message.

Answers

1.

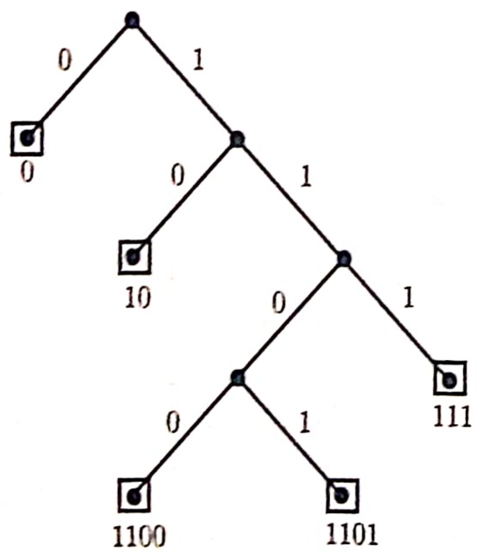
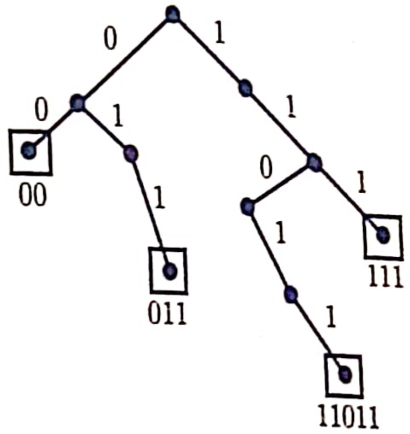


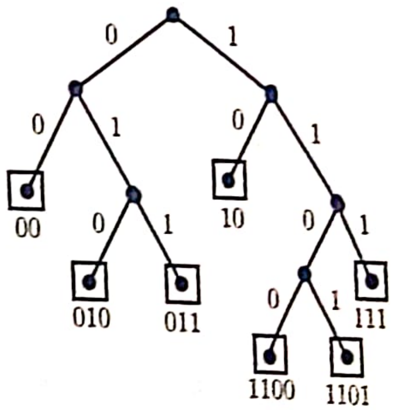
Figure 10.40

2. Sets A, C, D are prefix codes. Their binary trees are as shown below.



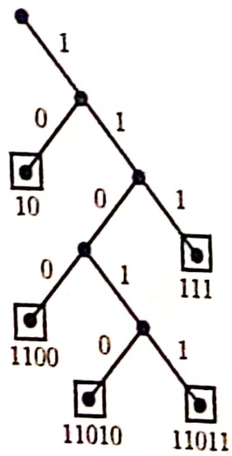
(A)

Figure 10.41



(C)

Figure 10.42



(D)

Figure 10.43

3. $P = \{00, 0100, 0101, 011, 10, 110, 111\}$

4. $W(T_1) = 46$, $W(T_2) = 45$, $W(T_3) = 36$,

5.

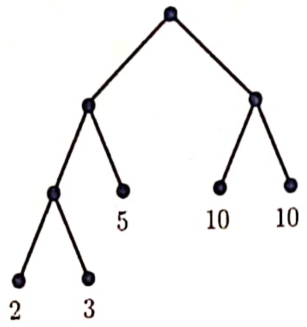


Figure 10.44

6.

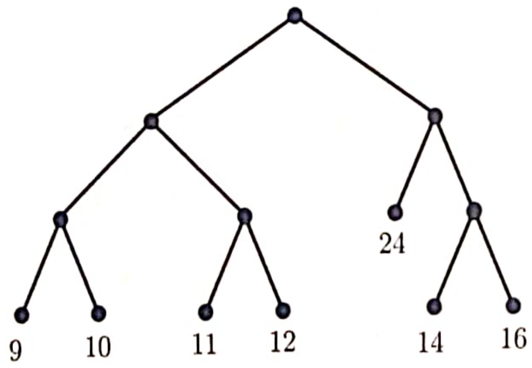


Figure 10.45

7. (1) $a : 010$, $b : 10$, $c : 0110$, $d : 110$, $e : 00$, $f : 111$, $g : 0111$

(2) $A : 00$, $B : 11011$, $C : 010$, $D : 111$, $E : 11010$, $F : 011$, $G : 10$, $H : 1100$

8. $A : 110$ $S : 111$ $C : 00$ $L : 01$ $U : 10$

code : 001100100100110111

9. $T : 010$ $K : 011$ $C : 100$ $R : 101$ $\square : 110$ $A : 111$ $E : 00$

code : 01011101100110100111110100

10. $R : 1000$ $S : 1001$ $L : 1010$ $T : 1011$ $D : 1100$ $\square : 1101$

$O : 000$ $A : 001$ $C : 010$ $E : 011$ $P : 111$

code : 1111000000111000100100110101101001010010011111101101111100

11. M: 0100 O: 0101 N: 0110 E: 0111

F: 1100 L: 1101 □: 1110 I: 1111

U: 000 C: 001 S: 10

code : 0100111110101111010101101110100000010010111101011000001101

12. O: 0110 T: 0111 H: 1100 E: 1101

W: 000 F: 001 A: 010 □: 111 L: 10

code: 00101010101011101100011110111110011011110000101010