Unit III

# 3 Graph Theory

## 3.1 : Basic Terminologies

**Q.1 Define graphs with examples.**

**Ans. : Graphs :** A graph is an ordered pair (V(G), E(G)) where

i) V(G) is non empty finite set of elements known as vertices or nodes. V(G) is called the vertex set.

ii) E(G) is a family of unordered pairs (not necessarily distinct) of elements of v, known as edges or arc or branches of G. E(G) is known as edge set.

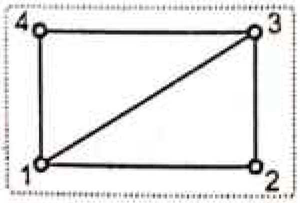
Graphs are so named because they can be represented diagrammatically in the plane.

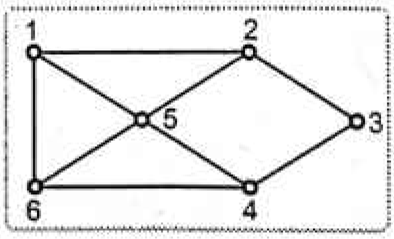
It is denoted by V(G, E).

a) Each vertex of G is represented by a point or small circle in the plane. In practical examples vertex set may be the set of states or cities or objects etc.

b) Every edge is represented by a continuous curve or straight line segment. Edges may be the route among states or cities or relation among objects etc. Diagrams of road maps, electrical circuits, chemical compounds, job scheduling family trees, all have two objects common namely vertices and edges.

Let us consider the following examples of graphs with V(G) and E(G).

1) 

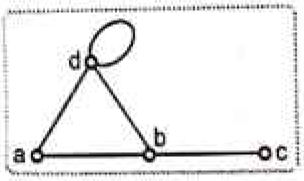

2) 


$V(G_1) = \{1, 2, 3, 4\}$ $V(G_2) = \{1, 2, 3, 4, 5, 6\}$

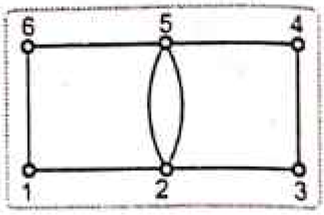
$E(G_1) = \{(1, 2), (1, 3)\ (1, 4)$ $(2, 3), (3, 4)\}$

$E(G_2) = \{(1, 2), (1, 5), (1, 6),$ $(2, 5), (2, 3)\ (3, 4), (4, 5),$ $(4, 6), (5, 6)\}$

3)




4)



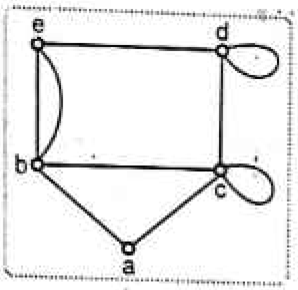

$V(G_3) = \{a, b, c, d\}$

$E(G_3) = \{(a, b), (a, d)\ (b, c)$ $(b, d), (d, d)\}$

$V(G_2) = \{1, 2, 3, 4, 5, 6\}$

$E(G_4) = \{(1, 2), (1, 6), (2, 3),$ $(2,5), (2, 5), (3, 4),$ $(4, 5), (5, 6)\}$

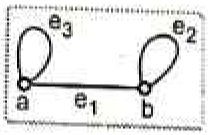
5)




$V(G_5) = \{a, b, c, d\}$

$E(G_5) = \{(a, b), (a, c), (b, e)\ (b, e), (b, c),$ $(c, c)\ (c, d)\ (d, d), (d, e)\}$

6)




$V(G_6) = \{a, b\}$

$E(G_6) = \{a, b)\ (a, a)\ (b, b) = \{e_1, e_2, e_3\}$

i) If x and y are two vertices of a graph G and unordered pair $\{x, y\} = (x, y) = e$ is an edge then we say that edge e joins x and y or e is incident to both vertices x and y.

In this case, vertices x and y are said to be incident one e.g. In example (1), e = (2, 3) $\therefore$ e is incident at 2 and 3 and vertices 2, 3 are one incident on e = (2, 3).

ii) Two vertices x and y are said to be adjacent to each other if the pair (x, y) is an edge of G.

If $e = (x, y)$ is an edge of G then x and y are said to be end vertices of e and we can say that e is incident at x and y.

iii) Two edges $e_1$ and $e_2$ are said to be adjacent if they have a common vertex i.e. If $e_1$ and $e_2$ are adjacent then $e_1 = \{x, y\}$ and $e_2 = \{y, z\}$.

iv) An edge joining a vertex to itself is called a loop. E.g. In example (5) there are 2 loops (c, c) and (d, d).

v) A pair of vertices of a graph is joined by two or more edges, such edges are called as multiple or parallel edges.

In example (4) (2, 5) (2, 5) are multiple edges.

## 3.2 : Types of Graphs

**Q.2 Explain different types of graphs.**

**1) Multigraph :** A graph in which a pair of vertices is joined by two or more edges is called a multigraph or multiple graph.

i.e. A graph having multiple edges is called a multigraph. In examples (1), (2), (3), (6), graphs are not multigraphs and graphs in examples (4), (5) are multigraphs.

**2) Pseudograph :** A graph having loops but no multiple edges is called a Pseudograph.

Graphs in examples (3), (5) and (6) are pseudographs. A graph having only loops is called a Haary graph. For example :

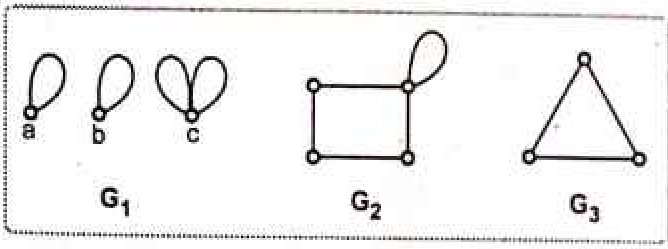
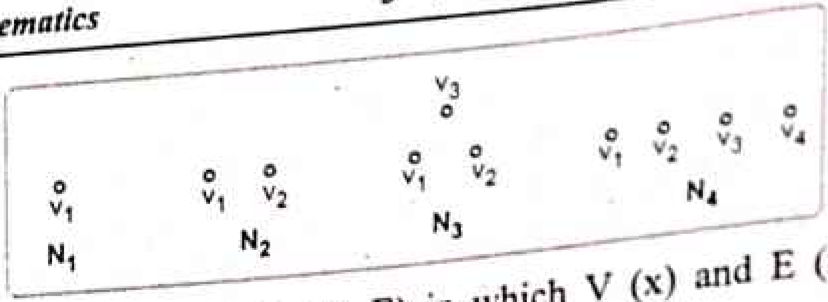



Fig. Q.2.1

Graph G is a pseudograph is well as Haary graph. Graph $G_2$ is Pseudograph but not Haary graph. Graph $G_3$ is neither Pseudo not Haary graphs.

**3) Simple graph :** A graph without loops and multiple edges is called a simple graph. Graphs in examples (1) and (2) are simple graphs. Graphs in examples (3), (4), (5), (6) are not simple graphs.

**4) Null graph :** A graph G (V, E) is said to be null graph if E is a empty set. Null graph on n vertices is denoted by $N_n$.




5) **Finite graph :** A graph G (V, E) in which V (x) and E (x) are finite sets is called a finite graph. Otherwise infinite graph.

6) **Directed graph :** A graph G (V, E) is said to be directed graph if the elements of E are an ordered pairs of vertices. E.g.

$E = \{(a, b), (b, c), (a, c)\}$

Here $(a, c) \neq (c, a)$, $(c, a) \notin E(G)$.

A graph which is not directed is called Non-directed graph or graph.

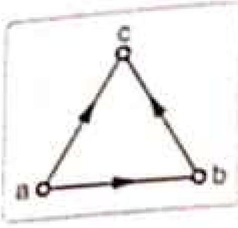



Fig. Q.2.2 Directed graph

7) **Weighted graph :** A graph G (V, E) in which some weight is assigned to every edge of G, is called weighted graph.

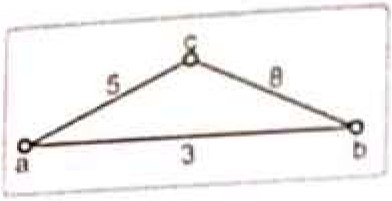



Fig. Q.2.3 Weighted graph

8) **Degree of a vertex :**

a) In a directed graph G the number of edges ending at vertex v is called the indegree of v. It is denoted by deg $G^{+}(v)$ or $d^{+}(v)$

b) **Outdegree :** In a directed graph G, the number of edges beginning at vertex v is called the outdegree of v. It is denoted by deg $G^{-}(v)$ or $d^{-}(v)$.

c) The number of edges incident at a vertex v of a graph G with self loops counted twice is called the degree of the vertex v. It is denoted by d(v). A vertex of degree one is called pendent vertex. A vertex of degree zero is called isolated vertex. An edge incident at pendent vertex is called pendent edge.

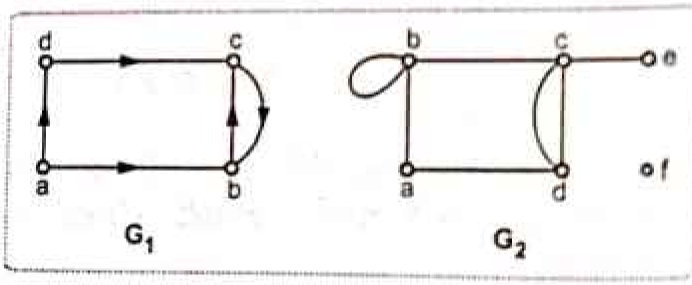



Fig. Q.2.4

e.g.

In graph $G_1$,

| Vertices | Indegree | Outdegree |
|---|---|---|
| a | 0 | 2 |
| b | 2 | 1 |
| c | 2 | 1 |
| d | 1 | 1 |

In graph $G_2$,

$$d(a) = 2 \ , \ d(b) = 2 + 2 = 4 \ , \ d(c) = 4 \ , \ d(e) = 1$$

$$d(d) = 3 \ , \ d(f) = 0$$

$\therefore$ f is an isolated vertex. e is a pendent vertex. An edge {c, e} is a pendent edge.

9) **Order and size of graph :** The number of vertices in a finite graph G is called the order of G. The number of edges in a finite graph G is called size of the graph. A graph of order n and size m is called (n, m) graph.

If G is a (p, q) graph then G has p vertices and q edges.

10) **Degree sequence of a graph :**

Let G be a graph with vertex set $V = \{v_1, v_2, v_3, \ldots v_n\}$ and $d_i = \deg(v_i)$ then the sequence $(d_1, d_2, d_3, \ldots d_n)$ in any order is called the degree sequence of G.

**Note :** 1) Vertices of G are ordered so that degree sequence is monotonically increasing.

2) Two graphs with same degree sequence are called to be degree equivalent. e.g.

$d(v_1) = 4$, $d(v_2) = 3$,

$d(v_3) = 2$, $d(v_5) = 5$,

$d(v_6) = 3)$, $d(v_4) = 1$

$\therefore$ Its degree seq. is (4, 3, 2, 1, 5, 3)

By relabelling vertices we may write degree sequence as (1, 2, 3, 3, 4, 5).

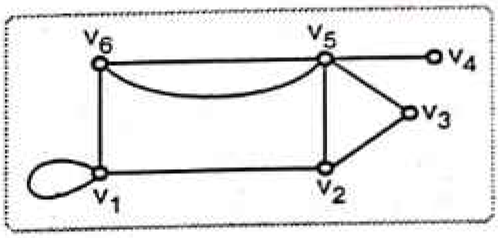



**Fig. Q.2.5**

**Q.3 Handshaking lemma : Let G (V, E) be any graph then $\sum_{v \in V} d(v) = 2q$ where q denotes the number of edges of G.**

**Ans. : Proof :** Let us argue by induction on q. Suppose q = 0 i.e. G has no edge i.e. E is an empty set. So $d(v) = 0, \forall v \in V$.

$\therefore \sum d(v) = 2q = 0$.

Let G be a graph with q > 0 edges. Choose any edge e = {u, v} of G. Consider the graph $G_1$ obtained from G as follows :

i) The vertex set of $G_1$ is same as the vertex set of G i.e.
$V(G_1) = V(G) = V = p$

ii) The edges of $G_1$ are all edges of G except e.

In other words, $G_1$ is obtained from G by deleting the edge e.

$\therefore$x is a (p, q − 1) graph.

$\therefore$ By induction principle, result is true for q − 1 edges

i.e. $$\sum_{x \in V} d(x) = 2(q-1) \qquad \dots \text{(Q.3.1)}$$

The degree of a vertex x other than u or v in $G_1$ is same as that at G. And the degree of u in $G_1$ is one less than the degree of u in G.

i.e. $$d_{G_1}(u) = d_G(u) - 1$$

Similarly $$d_{G_1}(v) = d_G(v) - 1$$

Hence equation (Q.3.1) becomes

$$\sum_{\substack{x \in V \\ x \neq u, v}} d(x) + d_{G_1}(u) + d_{G_1}(v) = 2(q-1)$$

$$\sum_{\substack{x \in V \\ x \neq u, v}} d(x) + d_{G_1}(u) - 1 + d_G(v) - 1 = 2(q-2)$$

$$\therefore \quad \sum_{x \in V} d(x) = 2q \quad \text{Hence the proof.}$$

The result is so named because it implies that if several people shake hands, the total number of hands shaken must be even as two hands are involved in one handshake.

**Note :** If $\sum_{v \in V} d(v) =$ Odd number then there does not exist any graph with this degree sequence.

**Q.4 Explain matrix representation of a graph with suitable examples.**

**[SPPU : Dec.-09, 10, May-10]**

**Ans. : 1. Adjacency Matrix :** Let G be a graph with n vertices and no parallel edges. The adjacency matrix of x is denoted by

$$A(G) = [a_{ij}]_{n \times n} \text{ and defined as}$$

$$a_{ij} = 1 \text{ if } v_i \text{ and } v_j \text{ are adjacent}$$

$$= 0 \text{ if } v_i \text{ and } v_j \text{ are not adjacent.}$$

**Note :** i) A (G) is asymmetric - binary matrix.

ii) The principal diagonal entries are all zeros if G has no loops.

iii) The $i^{th}$ row sum = $i^{th}$ column sum = $d(v_i)$

e.g. 1) The adjacent matrices of the following graphs are

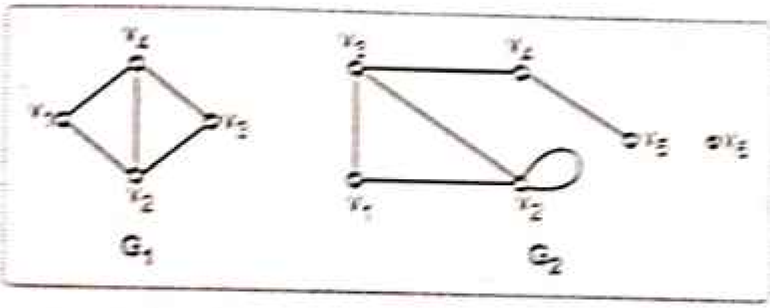



Fig. Q.4.1

$$A(G_1) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}_{4 \times 4} \end{array}, \quad A(G_2) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{6 \times 6} \end{array}$$

The adjacency matrix for a multigraph G is a $n \times n$ Matrix $A(G) = \left[a_{ij}\right]_{n \times n}$ where

$$a_{ij} = \text{Number of edges joining } v_i \text{ and } v_j$$

The adjacency matrix of the following graph is

$$A(G) = \begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & 1 & 1 & 0 & 0 & 0 \\ v_2 & 1 & 0 & 2 & 0 & 1 \\ v_3 & 0 & 2 & 0 & 1 & 0 \\ v_4 & 0 & 0 & 1 & 0 & 2 \\ v_5 & 0 & 1 & 0 & 2 & 1 \end{array}$$

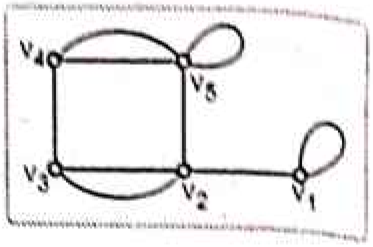



**Fig. Q.4.2**

**2. Incidence Matrix :** Let G be a graph with n vertices and m edges without self loops. The incidence matrix is denoted by X (G) or I (G) and defined as

$$X(G) = [x_{ij}]_{n \times m} \text{ where}$$

$$x_{ij} = 1 \text{ if } j^{th} \text{ edge is incident on } i^{th} \text{ vertex } v_i.$$

$$= 0 \text{ otherwise.}$$

X (G) is a $n \times m$ matrix whose n rows correspond to the n vertices and m columns correspond to m edges. The graph and it's incidence matrix are given below

$$X(G) = \begin{array}{c|ccccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\ \hline v_1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ v_2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v_3 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ v_4 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ v_5 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}_{5 \times 7}$$

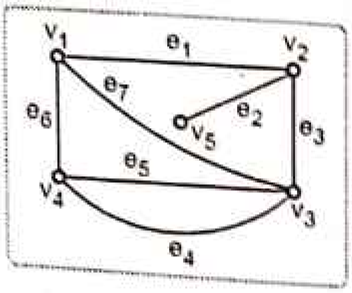



**Fig. Q.4.3**

**Properties of Incidence Matrix**

1) It contains only 0 and 1.
2) Each column in the incidence matrix has exactly two 1's appearing in that column.
3) The sum of elements in a row is equal to the degree of the corresponding vertex.
4) Two identical columns correspond to the parallel edges in graph.
5) A row with all 0's represents an isolated vertex.
6) A row with single 1 represents a pendent vertex.

The incidence matrix of a graph with loop is given as follows :

$$X(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}_{3\times 3} \end{array}$$

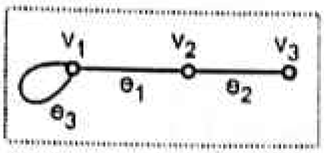


Fig. Q.4.4

### 3. Adjacency Matrix of a Diagraph (Directed graph)

Let G be a directed graph with n vertices and without parallel edges. The adjacency matrix is denoted by

Where, $A(D) = [a_{ij}]_{n\times n}$

$a_{ij} = 1$ if there is an edge directed from $v_i$ to $v_j$

$= 0$ otherwise

In network flow, adjacency matrix is also known as connection matrix or transition matrix. The adjancency matrix and diagraph are given below.

$$A(D) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{6\times 6} \end{array}$$

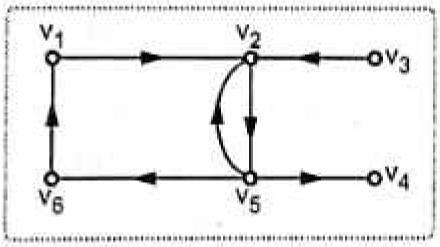


Fig. Q.4.5

**4. Incidence Matrix of Diagraph :** The incidence matrix of a diagraph with n vertices m edges and no self loops is a matrix.

$X(G) = [x_{ij}]_{n\times m}$ where

$x_{ij} = 1$ if $j^{th}$ edge $e_j$ is incident out of $i^{th}$ vertex $v_i$.

$= -1$ if $j^{th}$ edge $e_j$ is incident into of vertex $v_i$.

$= 0$ otherwise.

A graph and its incidence matrix are given below :

$$X(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}_{6\times 6} \end{array}$$

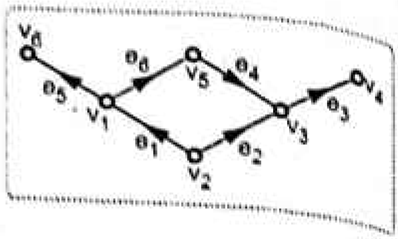



Fig. Q.4.6

**Note :** i) Sum of elements in each column of incidence matrix is zero.

**Q.5 Show that in a graph the number of vertices of odd degree is even.**

**Ans. :** Let G be a (p − q) graph.

By handshaking lemma $\sum_{v_i \in V} d(v_i) = 2q$

Now separate out vertices of even degree and odd degree

$$\therefore \quad \sum_{v_i \in V} d(v_i) = \sum_{\substack{x \in V \\ \text{even degree}}} d(x) + \sum_{\substack{y \in V \\ \text{odd degree}}} d(y) = 2q$$

$$\therefore \quad \sum_{\substack{v_i \in V \\ \text{odd degree}}} d(v_i) = 2q - \sum_{\substack{x \in V \\ \text{even degree}}} d(x) = \text{Even number}$$

∴ The sum of vertices of odd degree is even.

Hence the number of vertices of odd degree is even.

**Q.6 Show that the maximum number of edges in a simple graph with n vertices is $\frac{n(n-1)}{2}$.**

**Ans. :** Let G be a graph with n vertices m edges

∴ By handshaking lemma

$$\sum_{v \in V} d(v) = 2m \rightarrow$$

Let $x \in V$ $\therefore x$ must be adjacent to remaining $(x - 1)$ vertices

$\therefore d(x) = n - 1.\ \forall x \in V$

$\therefore$ Equation (1) $\Rightarrow (n - 1) + (n - 1) + \ldots$ n times $= 2$ m

$$n(n-1) = 2m$$

$$m = \frac{n(n-1)}{2}$$

Hence the maximum number of edges in any simple graph with n vertices is $\frac{n(n-1)}{2}$

**Q.7 Determine the number of edges in a graph with 6 nodes, 2 of degree 4 and 4 of degree 2. Draw to such graphs.** **☞[SPPU : Dec.-09]**

**Ans. :** Let G be the required graph with 6 nodes and m edges.

$\therefore$ By handshaking lemma

$$\sum_{v \in G} d(v) = 2m$$

$$d(v_1) + d(v_2) + d(v_3) + d(v_4) + d(v_5) + l(v_6) = 2m$$

$$4 + 4 + 2 + 2 + 2 + 2 = 2m$$

$$2m = 16$$

$$m = 8$$

Hence 8 edges are required.

Two such graphs are given below.

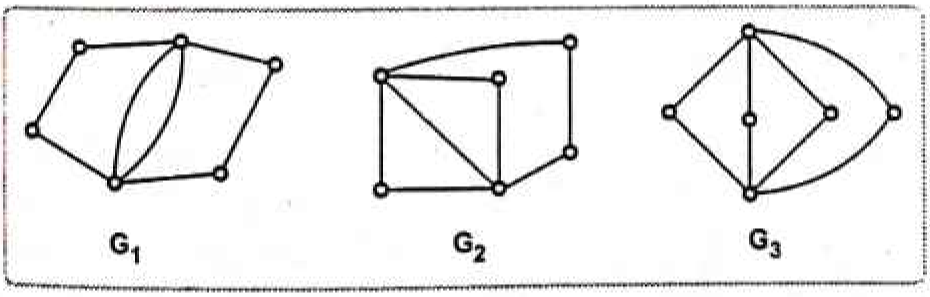



**Fig. Q.7.1**

**Q.8 Is it possible to construct a graph with 12 nodes such that 2 of the nodes have degree 3 and the remaining have degree 4.**

**[SPPU : Dec.-10]**

**Ans. :** Let G be the required graph with 12 vertices.

By handshaking lemma.

$$\sum_{v \in V(G)} d(v) = 2m$$

$$(3+3)+(4+4+4+4+4+4+4+4+4+4) = 2m$$

$$6+40 = 2m$$

$$\Rightarrow \quad m = 23$$

$\therefore$ It is possible to construct such graph.

**Q.9 Is graph exist for the degree sequence 4, 4, 3, 3, 2, 2, 1.**

**Ans. :** Now apply handshaking lemma

$$\sum d(v) = 2m = \text{Even}$$

$$4+4+3+3+2+2+1 = \text{Even}$$

$$19 = \text{Even which is impossible}$$

$\therefore$ Such graph does not exist.

**Q.10 How many simple labelled graphs with n vertices are there ?**

**[SPPU : May-10]**

**Ans. :** We know that a simple graph with n vertices has maximum possible number of edges $\frac{n(n-1)}{2} = m$ (say).

To construct a simple graph with e edges and n vertices, can be done in $\binom{m}{e}$ ways.

i.e. ${}^{m}C_{e}$ ways where $m = \frac{n(n-1)}{2}$ and $0 \le e \le m$

Hence the total number of ways to construct such graphs is given by

$$\binom{m}{0}+\binom{m}{1}+\binom{m}{2}+\ldots+\binom{m}{m} = 2^m = 2^{\left(\frac{n(n-1)}{2}\right)} \text{ ways.}$$

**Q.11 Show that a simple graph of order 4 and size 7 does not exist.**

**Ans. :** Let G be a simple graph with 4 vertices.

Then G has at most $\frac{n(n-1)}{2} = \frac{4 \times 3}{2} = 6$ edges.

But given that G has 7 edges which is contradiction.

$\therefore$ there can not be a simple graph with 4 vertices and 7 edges.

**Q.12 Explain i) Regular graph ii) Complete graph iii) Bipartite graph iv) Complete bipartite graph.**

**Ans. : I) Regular Graph :** A graph G is said to be r-regular graph if every vertex of G has degree r.

i) Regular graph of degree zero is called null graph.

ii) A regular graph of degree 3 is called cubic graph.

**e.g.**

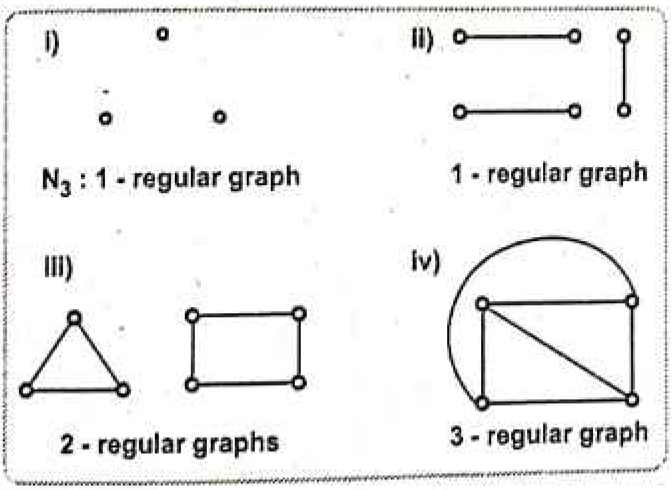



**Fig. Q.12.1**

**II) Complete Graph :** A simple graph G in which every pair of distinct vertices are adjacent is called a complete graph. If G is a complete graph on n vertices then it is denoted by $K_n$.

In a complete graph, there is an edge between every pair of distinct vertices.

In graph $K_n$, every vertex is adjacent to remaining $n-1$ vertices so degree of each vertex is $n-1$.

Thus $K_n$ is a $(n-1)$ - regular graph.

$K_n$ has exactly $\frac{n(n-1)}{2}$ edges.

Consider the following examples :

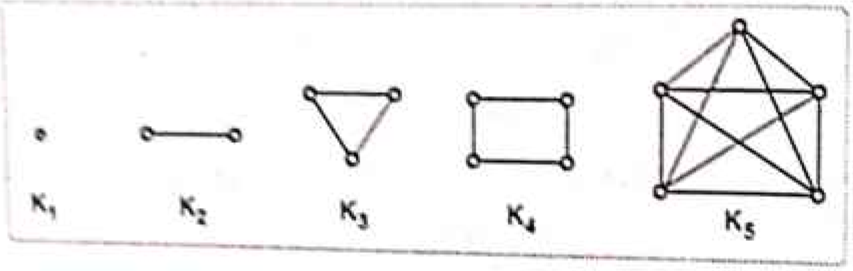



**Fig. Q.12.2**

**III) Bipartite Graph :** A graph G (v, E) is said to be bipartite graph if its vertex set can be partitioned into two disjoint subsets say $v_1$ and $v_2$ such that $v_1 \cup v_2 = v$ and $v_1 \cap v_2 = \phi$ and every edge of G joins a vertex of $v_1$ to a vertex of $v_2$. In Bipartite graph, vertices of $v_1$ should not be adjacent. It is free from loops.

Following graphs are bipartite graphs

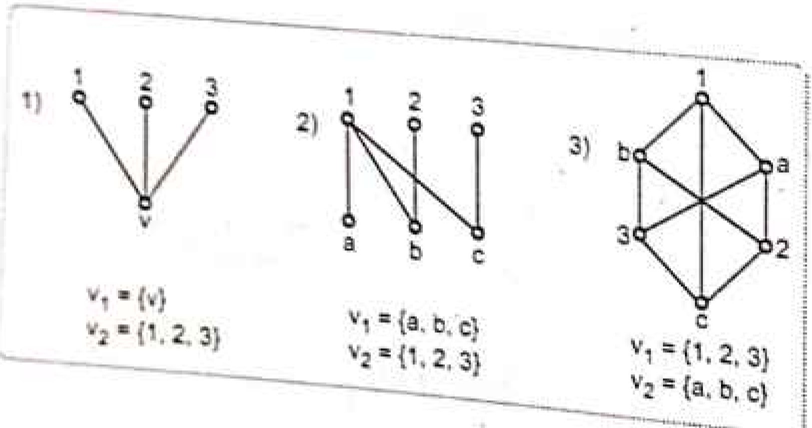



**Fig. Q.12.3**

**IV) Complete Bipartite graph :** A bipartite graph G (v, E), $v_1 \cup v_2 = v$ and $v_1 \cap v_2 = \phi$ is said to be complete Bipartite graph if each vertex of $V_i$ is joined to every vertex of $v_2$ by a unique edge.

If $|v_1| = m$, $|v_2| = n$, then the complete bipartite graph $G(v_1 \cup v_2, E)$ is denoted by $K_{m,n}$

Examples :

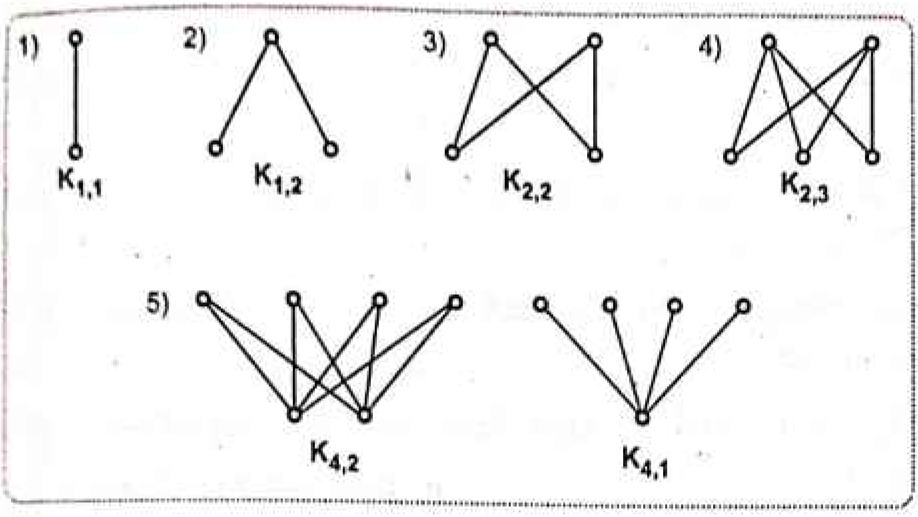



Fig. Q.12.4

The graph $K_{1,n}$ is called as star graph.

**Q.13 Is there exist any complete bipartite graph with 7 vertices and 14 edges ?**

**Ans. :** First find all possible bipartitions of 7. They are 6 + 1, 5 + 2, 4 + 3.

We know that, if $G(v_1 \cup v_2, E)$ is a bipartite graph then the number edges in G is equal to $|v_1| \cdot |v_2|$

i.e. $|E| = |v_1| \cdot |v_2|$

Here $|E| = 14$ But $6.1 = 6$, $5.2 = 10$, $4.3 = 12$

Therefore the complete bipartite graphs with 7 vertices has 6 or 10 or 12 edges only.

Therefore any complete bipartite graph with 7 vertices and 14 edges.

**Q.14 Explain isomorphism of graphs with examples.**

**[SPPU : Dec.-12]**

**Ans. :** In real life we come across so many similar objects or figures with respect to size, shape or orientation. Similarly there are a few concepts in graph theory which deal with the similarity of graphs w.r.t. number of vertices or number of edges, number of regions and so on. Among all such similarities the most important one is an isomorphism of graphs.

**Definition :** Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two graphs. $G_1$ and $G_2$ are said to be isomorphic graphs if

i) There exists a bijective function $\phi : V_1 \to V_2$

ii) There exists a bijective function $\psi : E_1 \to E_2$ such that $e = (x, y)$ is an edge in $G_1$ iff $(\phi(x), \phi(y))$ is an edge in $G_2$.

The pair of functions $\phi$ and $\psi$ is called an isomorphism of $G_1$ and $G_2$. It is denoted by $G_1 \equiv G_2$.

Suppose two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are isomorphic graphs. Then it is clear that

i) $|V_1| = |V_2|$ i.e. $G_1$ and $G_2$ must have same number of vertices.

ii) $|E_1| = |E_2|$ i.e. $G_1$ and $G_2$ must have same number of edges.

iii) $G_1$ and $G_2$ must have an equal number of vertices with the same degree.

iv) $G_1$ and $G_2$ must have an equal number of loops.

v) $G_1$ and $G_2$ must have same number of pendent.

vi) $G_1$ and $G_2$ must have same number of pendent edges.

vii) If u and v are adjacent in $G_1$ then the corresponding vertices in $G_2$ are also adjacent.

In general it is easier to prove two graphs are not isomorphic by proving that any one of the above property fails.

Consider the following example

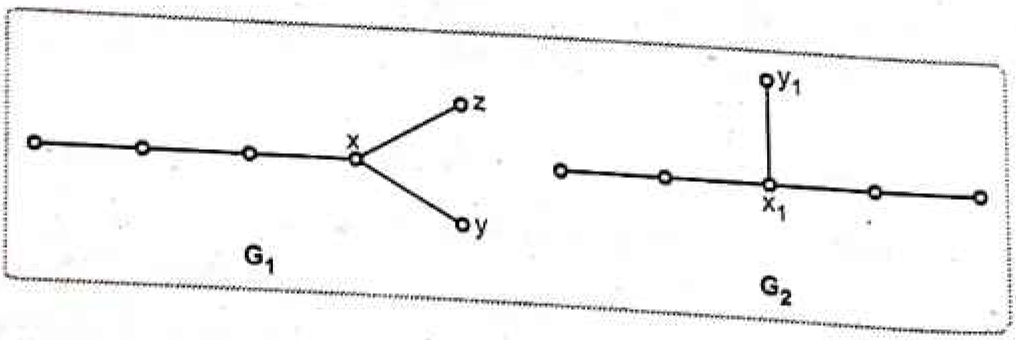



**Fig. Q.14.1**

These graphs have i) The same number of vertices.

ii) The same number of edges.

iii) An equal number of vertices of degree k.

These conditions are necessary for two graphs to be isomorphic but not sufficient.

In graph $G_1$, d (x) = 3, d (y) = 1, d (z) = 1 and x and y are adjacent to vertex x.

In graph $G_2$, d $(x_1)$ = 3, d $(y_1)$ = 1

there is only one pendent vertex adjacent to $x_1$. Hence adjacency is not preserved. Therefore $G_1$ is not isomorphic to $G_2$ i.e. $G_1 \not\equiv G_2$.

Note : Isomorphism of graphs is an equivalence relation.

**Examples :**

**Q.15 Draw all isomorphic graphs on vertices 2 and 3.**

Ans. : **i) For 3 vertices.**

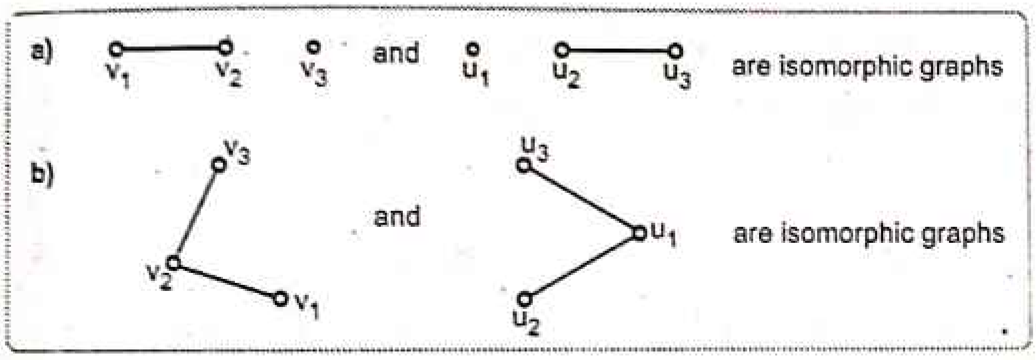



**Fig. Q.15.1**

**ii) For two vertices.**

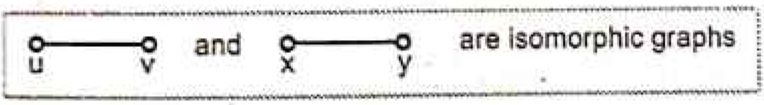



**Fig. Q.15.1 (a)**

**Q.16 Draw all non isomorphic graphs on 2 and 3 and 4 vertices.**

Ans. : All non-isomorphic graphs on 2 vertices are


**Fig. Q.16.1**

All non isomorphic graphs on 3 vertices.

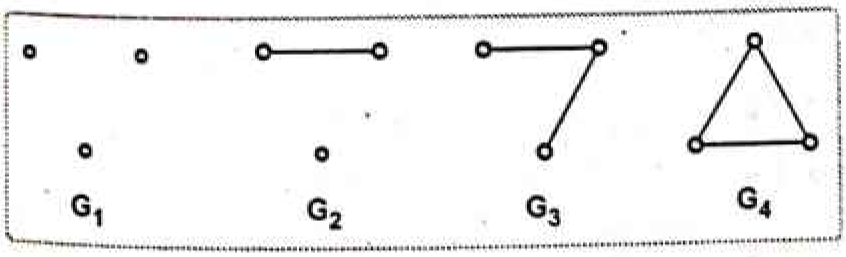



**Fig. Q.16.1 (a)**

All non isomorphic graphs on 4 vertices.

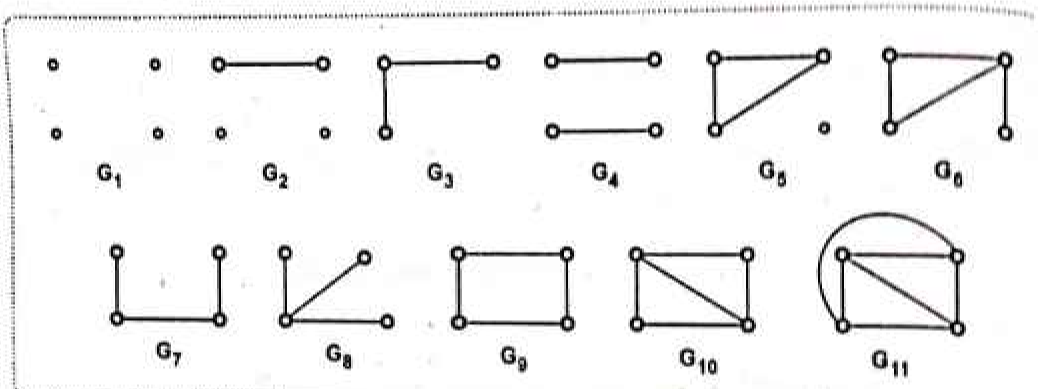



Fig. Q.16.1 (b)

**Q.17 Draw all non isomorphic graphs on 5 vertices and 5 edges.**

**Ans. :** The following are non isomorphic graphs with 5 vertices and 5 edges.

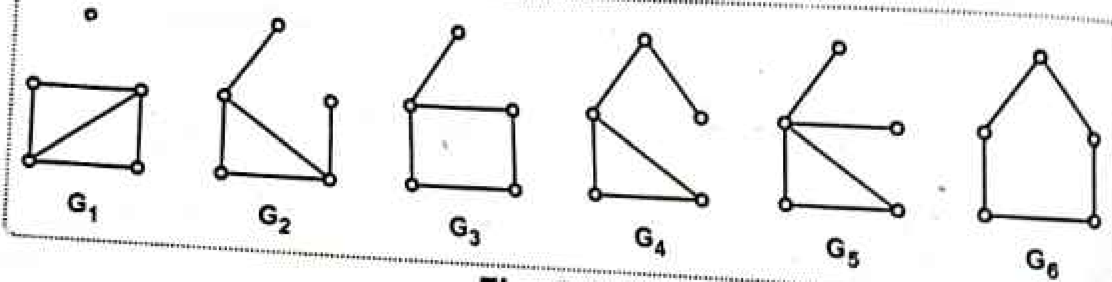



Fig. Q.17.1

**Q.18 Find whether the following pairs of graphs are isomorphic or not.**

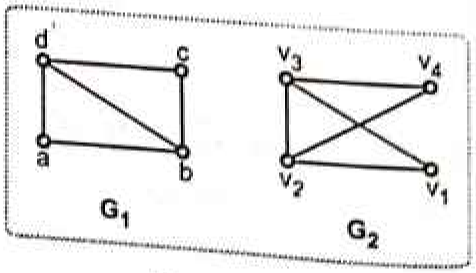



Fig. Q.18.1

**Ans. :** i) Both the graphs have 4 vertices and 5 edges.

Both have 2 vertices of degree 3 and 2 vertices of degree 2.

$\therefore \phi : \{a, b, c, d\} \to \{v_1, v_2, v_3, v_4\}$ such that

$a \to v_1$

$c \to v_4$

$b \to v_2$

$d \to v_3$

$\phi$ preserves adjacency and non-adjacency of vertices.

$\psi \to E_1 \to E_2$ Is bijective.

$\therefore$ $G_1$ is isomorphic to Graph $G_2$.

ii)

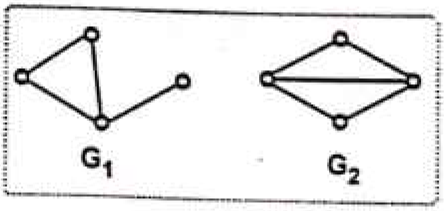



Fig. Q.18.1 (a)

As $G_1$ has 4 edges and $G_2$ has 5 edges, $G_1$ and $G_2$ are not isomorphic graphs.

iii) $G_1$ And $G_2$ are not isomorphic graphs because in $G_1$ vertices $v_1$ and $v_3$ of 4 degree are non adjacent while in $G_2$, the vertices x and y of degree 4 are adjacent.

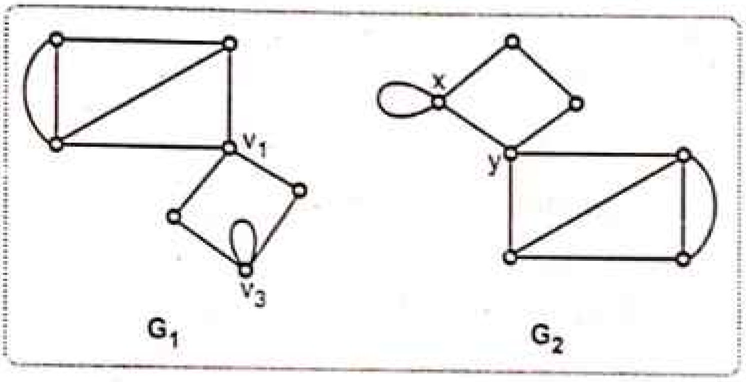



Fig. Q.18.1 (b)

**Q.19 Identify whether the given graphs are isomorphic or not.**

**[SPPU : Dec.-12]**

**Ans. :** In graph $G_1$, there are 2 vertices of degree 3. But in $G_2$, there is only one vertex of degree 3. So $G_1$ and $G_2$ are not isomorphic graphs.

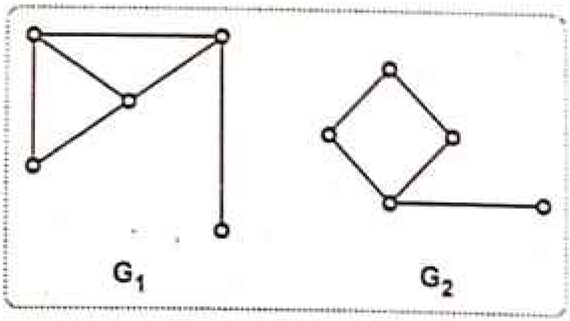



Fig. Q.19.1

ii)

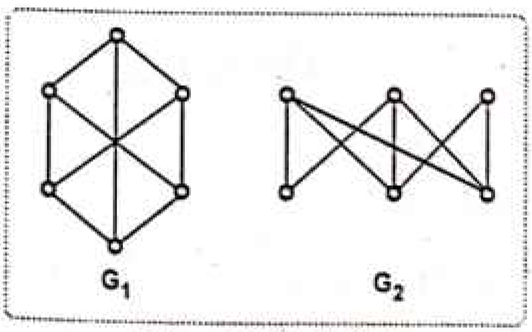



Fig. Q.19.1 (a)

Graph $G_1$ has 9 edges and $G_2$ has 8 edges.

Therefore $G_1$ and $G_2$ are not isomorphic graphs.

**Q.20 Show that the following graphs are isomorphic.**

[SPPU : May-14]

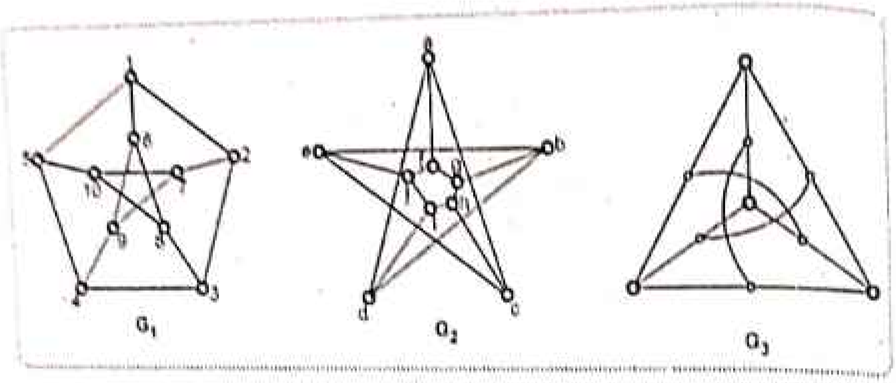


Fig. Q.20.1

**Ans. :** All graphs $G_1$, $G_2$ and $G_3$ have 10 vertices and 15 edges.
All these graphs are 3-regular graphs. Also they preserve adjacency. Hence all graphs are isomorphic. Isomorphism is given by

$1 \rightarrow f$ $2 \rightarrow g$ $3 \rightarrow h$ $4 \rightarrow i$ $5 \rightarrow j$ $6 \rightarrow a$

$7 \rightarrow b$ $8 \rightarrow c$ $9 \rightarrow d$ $10 \rightarrow e$

In the similar way, we can show that $G_1$ and $G_3$ are isomorphic graphs.

**Q.21 Are the graphs isomorphic ? Why ?**

**Ans. :** Given graphs $G_1$ and $G_2$ have 8 vertices and 10 edges.

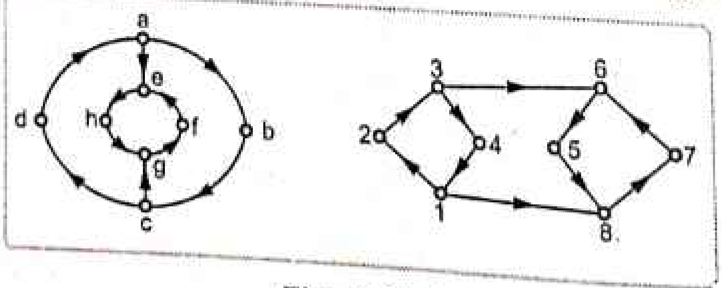


Fig. Q.21.1

Both the graphs have 4 vertices of degree 2 and 4 vertices of degree 3. Also the adjacency is preserved.

$\phi : v(G_1) \rightarrow v(G_2)$ is defined as

$a \rightarrow 1$, $b \rightarrow 2$, $c \rightarrow 3$, $d \rightarrow 4$, $e \rightarrow 8$, $f \rightarrow 5$, $g \rightarrow 6$, $h \rightarrow 7$,

$\phi$ is bijective.

$\therefore$ $G_1$ and $G_2$ are isomorphic graphs.

**Q.22 Explain how to obtain new graphs from old graphs with examples.**

**[SPPU : May-07, Dec.-09, 12]**

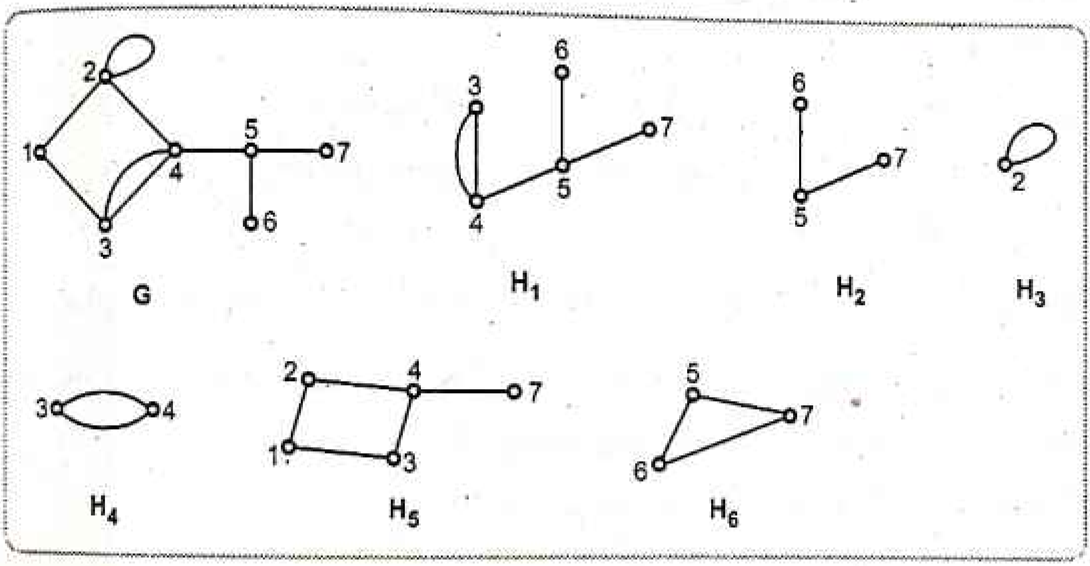
**Ans. :** A good mathematical theory must contain sufficient number of models and examples. Moreover it must have methods to generate new objects from old ones.

In this section we derive new graphs from old graphs.

1) **Subgraphs :** Let G (V, E) be any graph. A graph H $(V_1, E_1)$ is said to be subgraph of G if $V_1 \subseteq V$ and $E_1 \subseteq E$.

We also say that G is a supergraph of H.

e.g.




Graphs $H_1$, $H_2$, $H_3$ and $H_4$ are subgraphs of G. But graphs $H_5$ and $H_6$ are not subgraph as $(4, 7) \in E(H_5)$ but $(4, 7) \notin E(G)$ and $(6, 7) \in E(H_6)$ but $(6, 7) \notin E(G)$.

**Properties :**

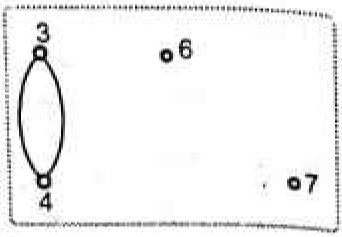
1) Each graph is a subgraph of itself.

2) A subgraph of a subgraph of a graph G is a subgraph of G.

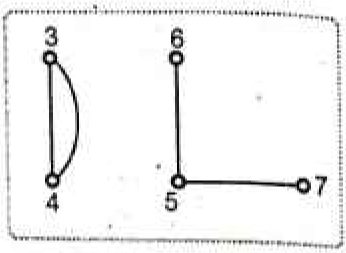
3) A graph G - {v} is a subgraph of G which is obtained from G by removing the vertex $v \in G$ and also the edges which are incident at v.

4) If $e \in (G)$ then G - e is a subgraph of G obtained from G by deleting the edge e.

In above example $H_1$ - {5} is




and $H_1$ - (4, 5) is given by




2) **Edge Disjoint Subgraphs :** Two subgraphs $H_1$ and $H_2$ of the graph G are said to be edge disjoint subgraphs of a graph G if there is no edge common between $H_1$ and $H_2$ i.e. $E(H_1) \cap E(H_2) = \phi$ It may have common vertex.

3) **Vertex Disjoint Subgraphs :** Two subgraphs $H_1$ and $H_2$ of the graph G are said to be vertex disjoint subgraphs if there is no vertex common between $H_1$ and $H_2$ i.e. $V(H_1) \cap V(H_2) = \phi$

**Note :** 1) All vertex disjoint subgraphs are edge disjoint subgraphs.

4) **Spanning Subgraph :** Let G (V, E) be any graph. A subgraph H of a graph G is said to be spanning subgraph if V (G) = V (H).

**Example :** Let G be the following graph :

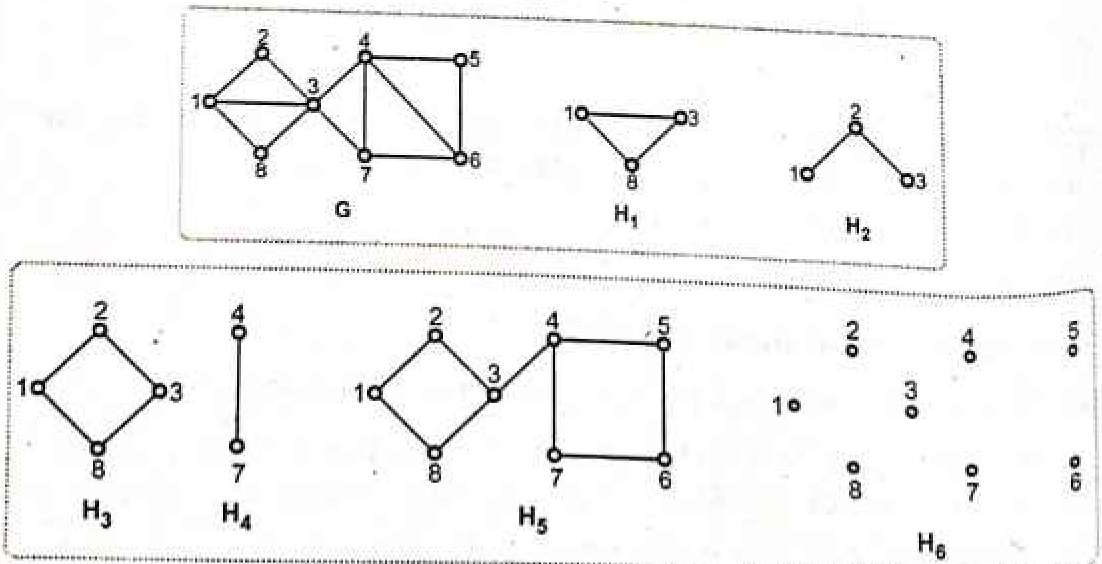



**Fig. Q.22.1**

Graphs $H_1, H_2, \ldots H_6$ are subgraphs of G.

$H_1$ and $H_2$ are edge disjoint subgraphs but not vertex disjoint subgraphs.

$H_3$ and $H_4$ are vertex disjoint subgraphs as well as edge disjoint subgraphs.

Subgraphs $H_5$ and $H_6$ are spanning subgraphs of G as

$V(H_5) = V(H_6) = V(G)$.

5) **Factors of a Graph :** Let G be any graph. A k-factor of a graph G is defined to be a spanning subgraph of the graph with the degree of each of its vertex being K. i.e. K-factor is a K-regular graph.

When G has a 1-factor, say $G_1$, if the number of vertices are even and edges of G are point disjoint.

In particular, $K_{2n+1}$ can not have a 1-factor but $K_{2n}$ can have 1-factor of graph.

**Example 1)**

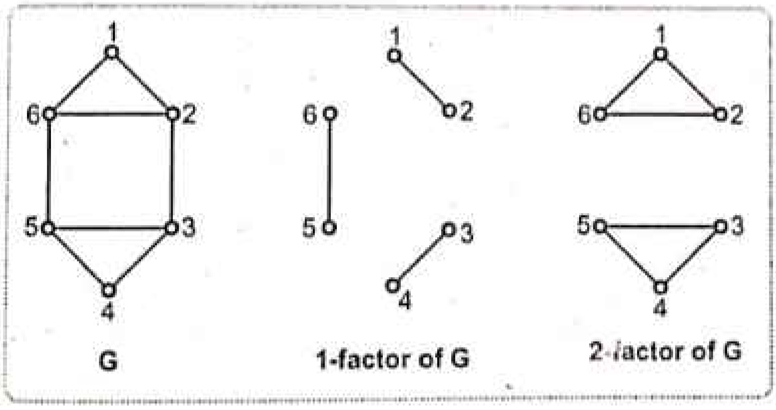



**Fig. Q.22.2**

**Example 2)**

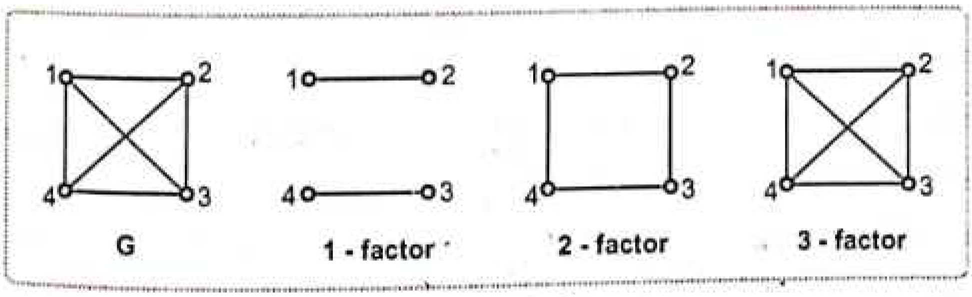



**Fig. Q.22.3**

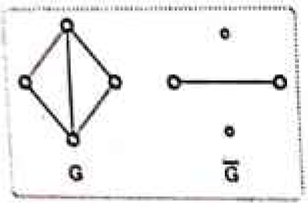
6) **Complement of a Graph :** Let G be a simple graph. The complement of G is denoted by $\overline{G}$ is the graph whose vertex set is the same as the

vertex set of G and in which two vertices are adjacent if and only if they are not adjacent in G.

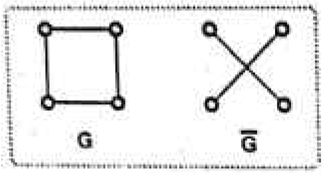
A graph is said to be self complementary graph if it is isomorphic to its complement.

e.g.

i)

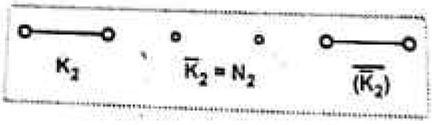



ii)

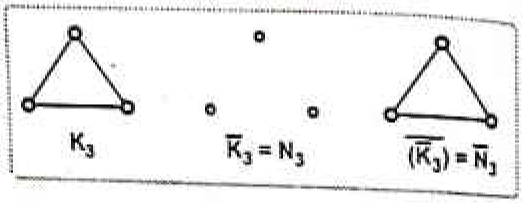



iii)

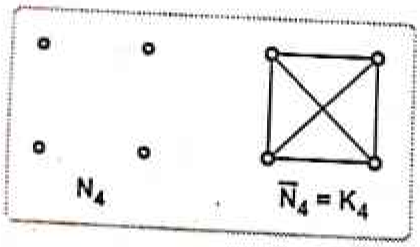

iv)



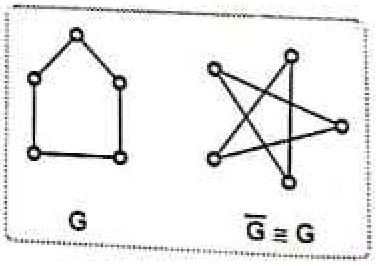

v)




vi)




vii)




G is isomorphic to $\overline{G}$. $\therefore$ G is self complementary graph.

**Note :**

1) For any graph G, $\overline{(\overline{G})} = G$

2) The complement of the null graph on n vertices is the complete graph $K_n$ on n vertices and vice versa.

3) $K_1$ is self complementary graph.

**Examples :**

**Q.23 For the following graphs, determine whether H (V', $E'$) is a subgraph of G where**

**i) $V' = \{A, B, C\}$, $E' = \{(A, B), (A, F)\}$**

ii) $V' = \{B, C, D\}$, $E' = \{(B, C), (B, D)\}$

iii) $V' = \{A, B, C, D\}$,

$E' = \{(A, C)\}$

[SPPU : May-07, Dec.09]

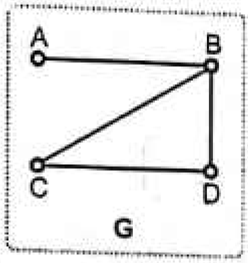



Fig. Q.23.1

**Ans. :** i) H is not a subgraph of G because $F \in V(H)$ but $F \notin V(G)$, so $V(H) \not\subset V(G)$

ii) Here $V' \subset V(G)$, $E' \subset E(G)$, so $H(V', E')$ is a subgraph of G.

iii) Here $V' \subset v(G)$, but $E' \not\subset E(G)$ Therefore $H(V', E')$ is not a subgraph of G.

**Q.24 Draw all self complementary graphs on 5 vertices.**

[SPPU : Dec.-12]

**Ans. :** The following graphs are self complementary graphs on 5 vertices.

Here $\overline{G_1} = G_2$ and $\overline{G_2} = G_1$

$\therefore G_1$ as well as $G_2$ are self complementary graphs.

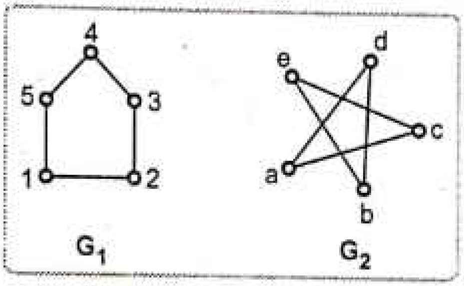



Fig. Q.24.1

**Q.25 Explain operations on graphs.**

**Ans. :** We define some standard operations of graphs like intersection, union, Ringsum etc.

**A) Intersection of Two Graphs :** The intersection of two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ is a graph $G(V, E)$ whose vertex set is $V = V_1 \cap V_2$ and edge set is $E = E_1 \cap E_2$. The intersection of $G_1$ and $G_2$ is denoted by $G_1 \cap G_2$.

e.g.

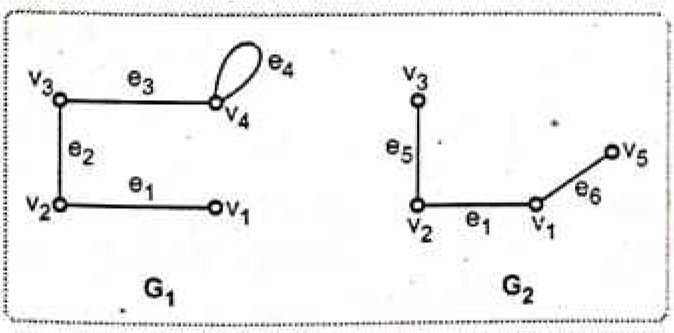



Fig. Q.25.1

$V_1 = \{v_1, v_2, v_3, v_4\}$ $V_2 = \{v_1, v_2, v_3, v_5\}$

$E_1 = \{e_1, e_2, e_3, e_4\}$ $E_2 = \{e_1, e_5, e_6\}$

Therefore $G = G_1 \cap G_2$ (v, E) where

$V = V_1 \cap V_2 = \{v_1, v_2, v_3\}$,

$E = E_1 \cap E_2 = \{e_1\}$

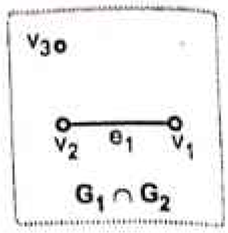



Fig. Q.25.2

**B) Union of Two Graphs :** Let $G_1(V_1, E_1)$, $G_2(V_2, E_2)$ be two graphs. The union of $G_1$ and $G_2$ is denoted by $G_1 \cup G_2 = G$ (v, E) and it is a graph whose vertex set is

$V = V_1 \cup V_2$ and Edge set is

$E = E_1 \cup E_2$

Consider the graphs $G_1$ and $G_2$ as shown in above example :

The union of $G_1$ and $G_2$ is given by G (v, E)

where $V = V_1 \cup V_2 = \{v_1, v_2, v_3, v_4, v_5\}$

$V = V_1 \cup V_2 = \{e_1, e_2, e_3, e_4, e_5, e_6\}$

**Note :** Both graphs $G_1$ and $G_2$ are subgraphs of $G_1 \cup G_2$.

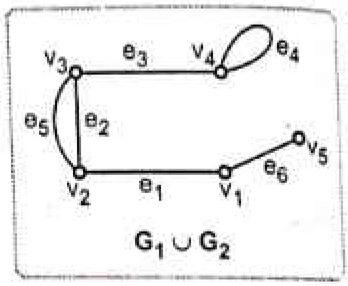



Fig. Q.25.3

**C) The Ring Sum of Two Graphs :** The ring sum of two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ is denoted by $G = G_1 \oplus G_2$ (V, E) whose vertex set is $V = V_1 \cup V_2$ and the edge set consists of those edges which are either in $E_1$ or in $E_2$ but not in both i.e. $E = (E_1 \cup E_2) - (E_1 \cap E_2)$.

The ring sum of above graphs $G_1$ and $G_2$ is given by $G(V, E) = G_1 \oplus G_2$

$$V = \{v_1, v_2, v_3, v_4, v_5\} = V_1 \cup V_2$$

$$E = (E_1 \cup E_2) - (E_1 \cap E_2)$$

$$= \{e_2, e_3, e_4, e_5, e_6\}$$

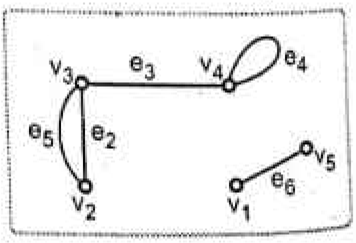



Fig. Q.25.4

**D) Sum of Two Graphs :** The sum of two vertex disjoint graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ is denoted by $G_1 + G_2 = G$ (V, E) is defined as the graph whose vertex set is $V(G_1 \cup G_2)$ and consisting of edges which are in $G_1$ or $G_2$ together with the edges obtained by joining each vertex of $G_1$ to each vertex of $G_2$. Thus $G_1 + G_2$ is nothing but the graph $G_1 \cup G_2$ in which each vertex of $G_1$ is joined to each vertex of $G_2$ by an edge.

e.g. If

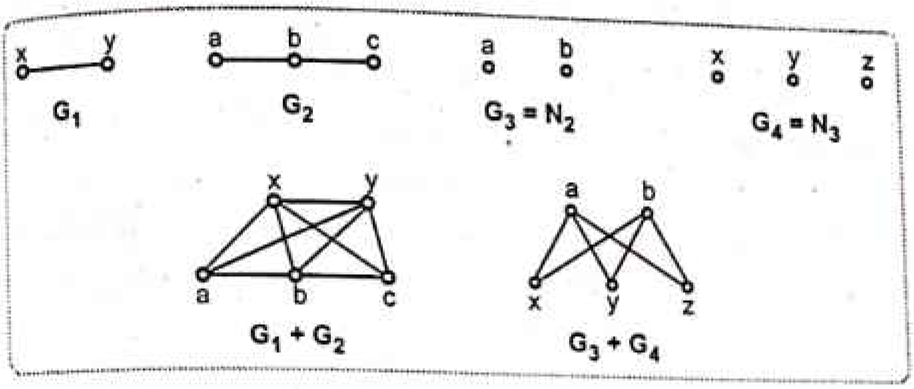



Fig. Q.25.5

Note : The sum $N_m + N_n$ of null graphs is nothing but the complete bipartite graph $K_{m,n}$.

**E) Product of Two Graphs :** Let $G_1 (V_1, E_1)$ and $G_2 (V_2, E_2)$ be two vertex disjoint graphs then the product of $G_1$ and $G_2$ is denoted by $G_1 \times G_2 = G (V, E)$ is a graph whose vertex set is $V = V_1 \times V_2$ and two edges $(x_1, x_2)$ and $(y_1, y_2)$ are adjacent if $x_1 = y_1$ and $x_2$ is adjacent to $y_2$ in $G_2$ or $x_2 = y_2$ and $x_1$ is adjacent to $y_1$ in $G_1$.

e.g. If

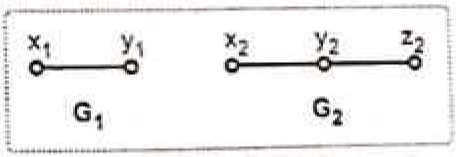



Fig. Q.25.6

Then $G_1 \times G_2$ is given below :

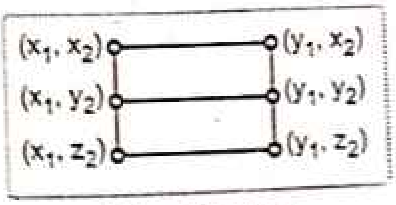



Fig. Q.25.7

**F) Decomposition :** A graph G is said to have been decomposed into two subgraphs H and K if $H \cup K = G$ and $H \cap K$ = Null graph i.e. each edge of G occurs either in H or in K but not in both. But vertices may occur in both. In this context isolated vertices are not considered.

e.g. The decomposition of G into H and K is given below :

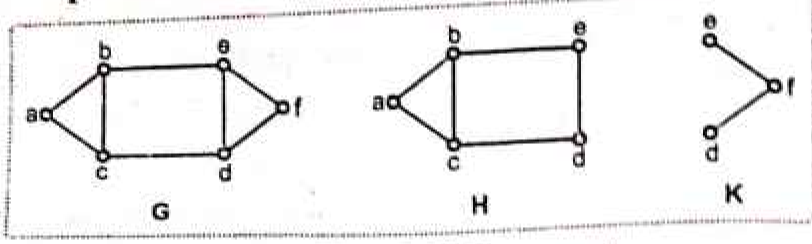



Fig. Q.25.8

**G) Fusion of vertices :** A pair of vertices a and b in a graph G are said to be fused if a and b are replaced by a single new vertex say c such that every edge that was incident on either a or b or both is incident on the new vertex c. The fusion of two vertices do not change the number of edges but reduced number of vertices by 1.

e.g.

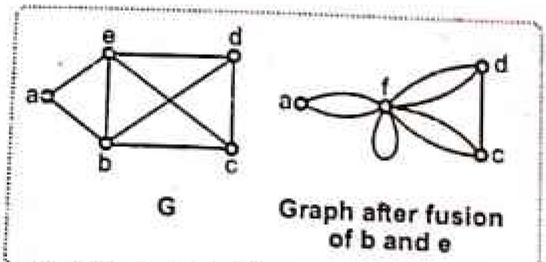



Fig. Q.25.9

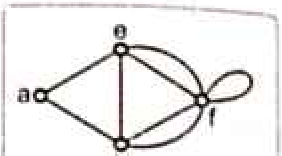



Fig. Q.25.10

## Q.26 Paths and circuits with examples.

**1) Path :** An alternating sequence of vertices and edges $v_0 - e_1 - e_2 - e_3 - \dots v_{n-1} - e_n - v_n$ beginning are ending with vertices in which each edge is incident with the two vertices immediately preceding if and following it is called a path.

The vertices $v_o$ and $v_n$ are called terminal vertices and $v_1, v_2, \dots v_{n-1}$ are called its interior vertices.

e.g. Let G be the following graph.

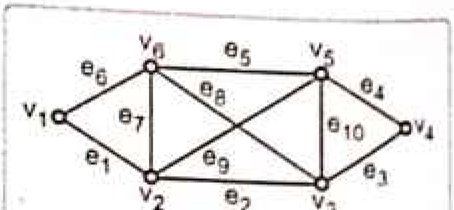



Fig. Q.26.1

Following are some examples of path

i) $v_1 - e_1 - v_2 - v_3$

ii) $v_2 - e_2 - v_3 - e_3 - v_4 - e_4 - v_5 - e_{10} - v_3 - e_2 - v_2$

iii) $v_6 - e_5 - v_5 - e_{10} - v_3 - e_8 - v_6$

iv) $v_1 - e_6 - v_6$

There are so many paths between every distinct pair of vertices of given graph G. Depending upon the nature of terminal vertices, there are two types at path.

A path in which terminal vertices are equal is called a **closed path.** A closed path is known as circuit. A path in which terminal vertices are distinct, is called an **open path.**

In above examples, paths in (i) and (iv) are open paths and (ii) and (iii) are closed paths.

**1. Simple Path :** A path in a graph G is said to be a simple path if the edges do not repeat in the path. Vertices may be repeated.

e.g. i) $v_1 - e_1 - v_2 - e_2 - v_3 - e_{10} - v_5$ is a simple path.

ii) $v_1 - e_1 - v_2 - e_2 - v_3 - e_{10} - v_5 - e_9 - v_2 - e_7 - v_6$ is a simple path in which $v_3$ is repeated.

iii) $v_3 - e_3 - v_4 - e_4 - v_5 - e_{10} - v_3 - e_8 - v_6$ is a simple path in which $v_3$ is repeated.

iv) $v_3 - e_3 - v_4 - e_4 - v_5 - e_{10} - v_3 - e_3 - v_4$ is not a simple path as an edge $e_3$ is repeated.

**2. Elementary Path :** A path in a graph G is said to be elementary path if vertices do not repeat in the path. Every elementary path is a simple path.

e.g. i) $v_1 - e_1 - v_2 - e_2 - v_3$ is an elementary path.

ii) $v_1 - e_1 - v_2 - e_2 - v_3 - e_8 - e_6 - e_7 - v_2$ is not an elementary path. But it is simple path.

**3. Simple and Elementary Circuits :** A closed path is known as circuit.

A simple path which is closed is called a simple circuit of graph.

In other words, A circuit in a graph G is said to be simple circuit if all edges of a circuit are distinct.

A circuit in a graph G is said to be elementary circuit if all vertices of a circuit are distinct except the terminal vertices i.e. the first and last vertices. The number of edges in any circuit (or path) is called the length of the circuit (or path).

In above graph G.

e.g. i) $v_1 - e_1 - v_2 - e_2 - v_3 - e_{10} - v_5 - e_9 - v_2 - e_1 - v_1$ is a circuit with $e_1$ repeated twice and $v_2$ is also repeated twice.

ii) $v_1 - e_1 - v_2 - e_2 - v_3 - e_{10} - v_5 - e_9 - v_2 - e_7 - v_6 - e_6 - v_1$ is a simple circuit but not elementary circuit as $v_2$ is repeated.

iii) $v_1 - e_1 - v_2 - e_7 - v_6 - e_6 - v_1$ is an elementary circuit.

**Q.27 Define connected and disconnected graphs.**

**Ans. :** A graph G is said to be connected graph if there exists a path between every pair of vertices. A graph which is not connected is called the disconnected graph.

A disconnected graph consists of two or more parts called components or blocks if each of which is connected and there is no path between two vertices if they belong to different components.

**Q.28 Explain edge and vertex connectivity.**

**Ans. : Edge Connectivity :** A set of edges of a connected graph G whose removal disconnects G is called a disconnecting set of G. A cutset is defined as a minimal disconnecting set i.e. A minimal set of edges whose removal from G gives a disconnected graph is called a cutset.

If a cutset of a graph contains only one edge, then that edge is called as an isthmus or bridge. The number of edges in the smallest cutset of G is called the edge connectivity of G. It is denoted by $\lambda$ (G).

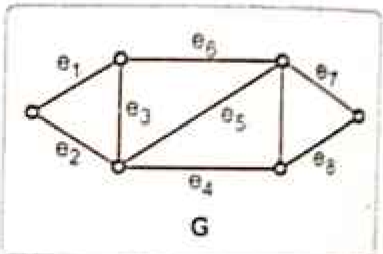



Fig. Q.28.1

e.g. Consider the following graph G.

Cutsets of G are as follows :

i) $\{e_4, e_5, e_6\}$, ii) $\{e_1, e_3, e_6\}$, iii) $\{e_1, e_2\}$,

A set $\{e_1, e_2, e_3\}$ is not a cutset because its subset $\{e_1, e_2\}$ is a cutset. The edge connectivity of graph G is 2. i.e. $\lambda(G) = 2$.

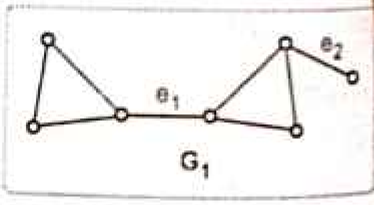



Fig. Q.28.2

Consider the following graph $G_1$.

Graph $G_1$ has edge connectivity 1 as $G_1 - \{e_1\}$ is a disconnected graph. $\therefore$ $e_1$ is an isthmus or Bridge.

$G - e_2$ is also disconnected graph. $\therefore$ $e_2$ is also isthmus.

**2) Vertex Connectivity :** The vertex connectivity k(G) of a simple connected graph G is defined as the smallest number of vertices whose removal disconnects the graph.

In graph G, the sets $\{v_2, v_5, v_4\}$, $\{v_2, v_3, v_4, v_5\}$, $\{v_2, v_3\}$ disconnect graph G. The smallest set is $\{v_2, v_3\}$

$\therefore$ k (G) = 2.

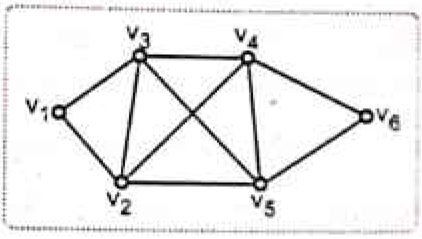




**Fig. Q.28.3**

1) A graph G is said to be k-connected if its vertex connectivity is k.

2) A graph G is said to be seperable graph if its vertex connectivity is one.

3) A vertex v of a connected graph G is said to be cut vertex if G – {v} is a disconnected graph.

4) $k(G) \leq \lambda(G) \leq \delta$

i.e. vertex connectivity ≤ edge connectivity ≤ minimum degree of a vertex in G and

$$\lambda(G) \leq \left[\frac{2e}{n}\right]$$

where e = Number of edges in G. n = Number of vertices in G.

**Q.29 Explain shortest path algorithm and Dijkstra's algorithm.**

**☞[SPPU : May-05, 07, 14, 15, Dec.-06, 07, 12, 13, 14, 15]**

**Ans. :** Suppose there is associated to each edge e of a graph a real number w (e). w (e) is called the weight of e. A weighted graph is a graph in which each edge has a weight. The weight of graph G is the sum of weight of all edges of G. Weighted graph has many applications in communication networks. Given a railway network connecting several cities, determine a shortest route between two cities.

We consider the weighted graph where the vertices are the towns, rail roads are the edges and the weight represent the distance between directly linked cities. Therefore weights are non negative integers. The problem is to find a path of minimum weight connectivity two given cities. Of course, this is possible theoretically. One has to list all paths, find their weights and select minimum one. But for large networks (large number of vertices and edges) this may not be efficient. So we required different method to take such problems. The algorithm was found by Dijkshtra (1959) and is known as Dijkshtra's algorithm.

A) Dijkshtra's algorithm to find the shortest path from the vertex a to vertex z of a graph G. Let G (v, E) be a simple graph and a, z ∈ V.

Suppose L(x) is the label of the vertex which represents the length of the shortest path from the vertex a. $W_{ij}$ = Weight of an edge $e_{ij} = (v_i, v_j)$

Consider the following steps :

**Step 1 :** Let P be the set of those vertices which have permanent labels and T = set of all vertices of G.

$$\text{Set } L(a) = 0,\ L(x) = \infty ;\ \forall\ x \in T \text{ and } x \neq a$$

$$P = \phi \text{ and } T = v.$$

**Step 2 :** Select the vertex v in T which has the smallest label. This label is called the permanent label of v. Also set P as $P \cup \{v\}$ and T as $T - \{v\}$.

If v = z then L (z) is the length of the shortest path from the vertex a to z and stop the procedure.

**Step 3 :** If $v \neq z$, then revise the labels of the vertices of T. i.e. The vertices which do not have permanent labels.

The new label of x in T is given by

$$L(x) = \min \{\text{old } L(x),\ L(v) + w(v, x)\}$$

where w (v, x) is the weight of the edge joining v and x. If there is no edge joining v and x then take $w(v, x) = \infty$.

**Step 4 :** Repeat the steps 2 and 3 until z gets the permanent label.

**Examples :**

**Q.30 Use Dijkstra's algorithm to find the shortest path between a and z.**

**[SPPU : May-05, 14, 8 Marks, Dec.-06, 6 Marks]**

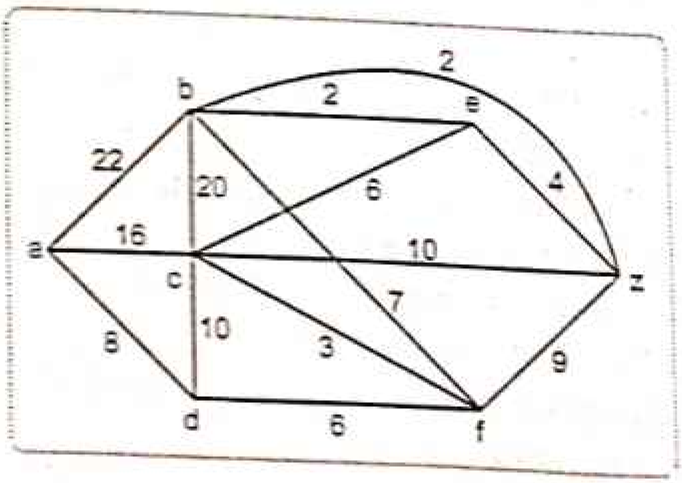



**Fig. Q.30.1**

**Ans. :**

**Step 1 :** $P = \varnothing, \quad T = \{a, b, c, d, e, f, z\}$

$L\{a\} = 0$

$L\{x\} = \infty, \quad \forall\ x \in T, \quad x \neq a$

**Step 2 :** $v = a$, the permanent label of a is 0

$P = \{a\}, \quad T = \{b, c, d, e, f, z\}$

$L\{b\} = \min\{\text{old } L(b), L(a) + w(a, b)\}$

$= \min\{\infty, 0 + 22\} = 22$

$L\{c\} = \min\{\infty, 0 + 16\} = 16$

$L\{d\} = \min\{\infty, 0 + 8\} = 8$

$L\{e\} = \min\{\infty, 0 + \infty\} = \infty$

$L\{f\} = \min\{\infty, 0 + \infty\} = \infty$

$L\{z\} = \min\{\infty, 0 + \infty\} = \infty$

$\therefore \quad L\{d\} = 8$ is the minimum label.

**Step 3 :** $v = d$, the permanent label of d is 8.

$P = \{a, d\}, \quad T = \{b, c, e, f, z\}$

$L\{b\} = \min\{\text{old } L(b), L(d) + w(d, b)\}$

$= \min\{22, 8 + \infty\} = 22$

$L\{c\} = \min\{16, 8 + 10\} = 16$

$L\{e\} = \min\{\infty, 8 + \infty\} = \infty$

$L\{f\} = \min\{\infty, 8 + 6\} = 14$

$L\{z\} = \min\{\infty, 8 + \infty\} = \infty$

$\therefore \quad L\{f\} = 14$ is the minimum label.

**Step 4 :** $v = f$, the permanent label of f is 14.

$P = \{a, d, f\}, \quad T = \{b, c, e, z\}$

$L\{b\} = \min\{\text{old } L(b), L(f) + w(b, f)\}$

$= \min\{22, 14 + 7\} = 21$

$L\{c\} = \min\{16, 14 + 3\} = 16$

$L\{e\} = \min\{\infty, 14 + \infty\} = \infty$

$L\{z\} = \min\{\infty, 14 + 9\} = 23$

$\therefore$ $L\{c\} = 16$ is the minimum label.

**Step 5 :** $v = c$, the permanent lable of c is 16.

$P = \{a, d, f, c\}$, $T = \{b, e, z\}$

$L\{b\} = \min\{\text{old } L(b), L(f) + w(f, b)\}$

$= \min\{21, 16 + 20\} = 21$

$L\{e\} = \min\{\infty, 16 + 6\} = 22$

$L\{z\} = \min\{23, 16 + 10\} = 23$

$\therefore$ $L\{b\} = 2$ is the minimum label.

**Step 6 :** $v = b$, the permanent label of b is 21.

$P = \{a, d, f, c, b\}$, $T = \{e, z\}$

$L\{e\} = \min\{\text{old } L(e), L(b) + w(e, b)\}$

$= \min\{22, 21 + 2\} = 22$

$L\{z\} = \min\{23, 21 + 2\} = 23$

$\therefore$ $L\{e\} = 22$ is the minimum label.

**Step 7 :** $v = e$, the permanent label of e is 22.

$P = \{a, d, f, c, b, e\}$, $T = \{z\}$

$L(z) = \min\{\text{old } L(z), L(e) + w(e, z)\}$

$= \min\{23, 22 + 4\} = 23$ which is the minimum label

**Step 8 :** $v = z$, the permanent label of z is 23.

Hence the length of the shortest path from a to z is 23.

The shortest path is adfz or adfbz.

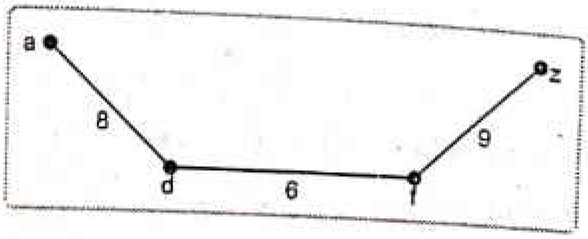



Fig. Q.30.1 (a)

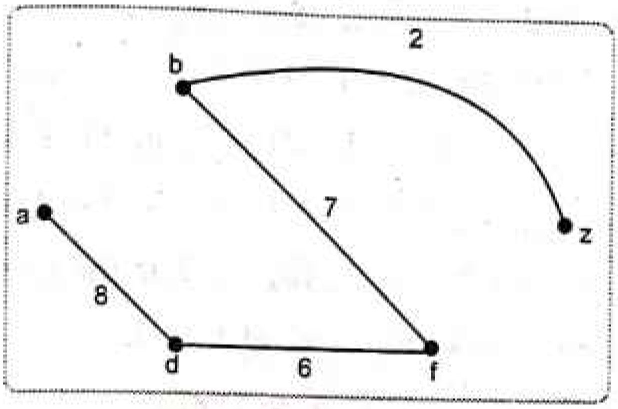



Fig. Q.30.1 (b)

**Q.31 Find the shortest path from a-z in the given graph using Dijkstra's algorithm.** [SPPU : May-07, Dec.-07, 15]

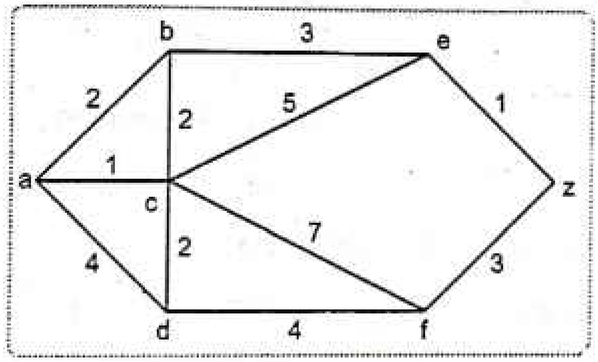



Fig. Q.31.1

**Ans. : Step 1 :**

Set $P = \varnothing$, $T = \{a, b, c, d, e, f, z\}$

$$L\{a\} = 0$$

$$L\{x\} = \infty, \ \forall x \in T, \ x \neq a$$

**Step 2 :** $v = a$, the permanent label of a is 0.

$$P = \{a\}, \quad T = \{b, c, d, e, f, z\}$$

$$L\{b\} = \min\{\text{old } L(b), L(a) + w(a, b)\}$$

$$= \min\{\infty, 0 + 2\} = 2$$

$L\{c\} = \min\{\infty, 0 + 1\} = 1$ $L\{d\} = \min\{\infty, 0 + 4\} = 4$

$L\{e\} = \min\{\infty, 0 + \infty\} = \infty$ $L\{f\} = \min\{\infty, 0 + \infty\} = \infty$

$L\{z\} = \min\{\infty, 0 + \infty\} = \infty$ $\therefore L\{c\} = 1$ is the minimum label.

**Step 3** : v = c, the permanent label of c is 1.

P = {a, c} , T = {b, d, e, f, z}

L {b} = min {2, 1 + 2} = 2 L {d} = min {4, 1 + 2} = 3

L {e} = min {∞, 1 + 5} = 6 L {f} = min {∞, 1 + 7} = 8

L {z} = min {∞, 1 + ∞} = ∞ ∴ L {b} = 2 is the minimum label.

**Step 4 :** v = b, the permanent label of b is 2.

P = {a, c, b} , T = {d, e, f, z}

L {d} = min {3, 2 + ∞} = 3 L {e} = min {6, 2 + 3} = 5

L {f} = min {8, 2 + ∞} = 8 L {z} = min {∞, 2 + ∞} = ∞

∴ L {d} = 3 is the minimum label.

**Step 5 :** v = d, the permanent label of d is 3.

P = {a, c, b, d}, T = {e, f, z}

L {e} = min {5, 3 + ∞} = 5 L {f} = min {8, 3 + 4} = 7

L {z} = min {∞, 3 + ∞} = ∞∴ L {e} = 5 is the minimum label.

**Step 6 :** v = e, the permanent label of e is 5.

P = {a, c, b, d, e}, T = {f, z}

L {f} = min {7, 5 + ∞} = 7 L {z} = min {∞, 5 + 1} = 6

∴ L {z} = 6 is the minimum label.

**Step 7 :** v = z, the permanent label of z is 6.

Hence the length of shortest path from a to z is 6.

The shortest path is a b e z.

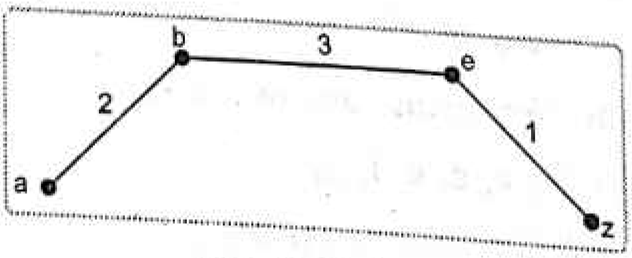



**Fig. Q.31.1 (a)**

**Q.32 Find the shortest path between a-z for the given graph : using Dijkstra's algorithm.**

**[SPPU : Dec.-12, 13, 14, May-15, 8 Marks]**

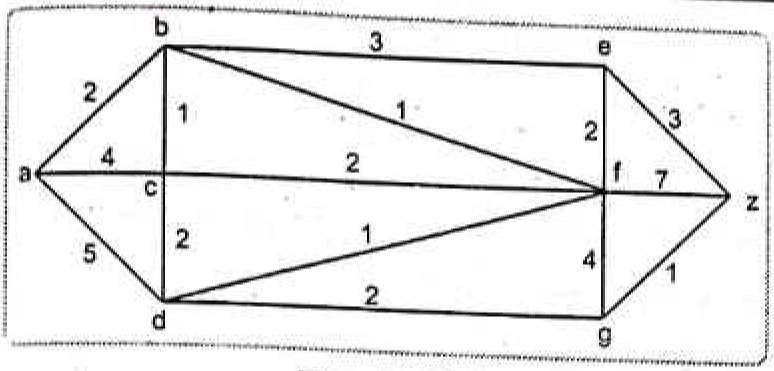



Fig. Q.32.1

**Ans. :**

**Step 1 :** Set $P = \varnothing$, $T = \{a, b, c, d, e, f, g, z\}$

$L\{a\} = 0$

$L\{x\} = \infty, \quad \forall\ x \in T, \quad x \neq a.$

**Step 2 :** $v = a$, the permanent label of a is 0.

$P = \{a\}, \quad T = \{b, c, d, e, f, g, z\}$

$L\{b\} = \min\{\text{old } L(b),\ L(a) + w(a, b)\} = \min\{\infty, 0 + 2\} = 2$

$L\{c\} = \min\{\infty, 0 + 4\} = 4 \qquad L\{d\} = \min\{\infty, 0 + 5\} = 5$

$L\{e\} = L\{f\} = L\{g\} = L\{z\} = \infty$

$\therefore$ $L\{b\} = 2$ is the minimum label. The permanent label of b is 2.

**Step 3 :** $v = b$

$P = \{a, b\}, \quad T = \{c, d, e, f, g, z\}$

$L\{c\} = \min\{L(c), L(b) + w\{b, c\}\} = \min\{4, 2 + 1\} = 3$

$L\{d\} = \min\{5, 2 + \infty\} = 5 \qquad L\{e\} = \min\{\infty, 2 + 3\} = 5$

$L\{f\} = \min\{\infty, 2 + 1\} = 3 \qquad L\{g\} = L\{z\} = \infty$

$\therefore$ $L\{c\} = L\{f\} = 3$ are the minimum labels.

**Step 4 :** $v = c$ or $f$

Let $v = f$, permanent label of f is 3.

$P = \{a, b, f\}, \quad T = \{c, d, e, g, z\}$

$L\{c\} = \min\{3, 3 + 2\} = 3 \qquad L\{d\} = \min\{5, 3 + 1\} = 4$

$L\{e\} = \min\{5, 3 + 2\} = 5 \qquad L\{g\} = \min\{\infty, 3 + 4\} = 7$

$L\{z\} = \min\{\infty, 3 + 7\} = 10$

$\therefore$ $L\{c\} = 3$ is the minimum label.

**Step 5 :** $v = c$, permanent label of $c = 3$.

$P = \{a, b, f, c\}$, $T = \{d, e, g, z\}$

$L\{d\} = \min\{4, 3 + 2\} = 4$ $\quad$ $L\{e\} = \min\{5, 3 + \infty\} = 5$

$L\{g\} = \min\{7, 3 + \infty\} = 7$ $\quad$ $L\{z\} = \min\{10, 3 + \infty\} = 10$

$\therefore$ $L\{d\} = 4$ is the minimum label.

**Step 6 :** $v = d$, permanent label of $d = 4$.

$P = \{a, b, f, c, d\}$, $T = \{e, g, z\}$

$L\{e\} = \min\{5, 4 + \infty\} = 5$ $\quad$ $L\{g\} = \min\{7, 4 + 2\} = 6$

$L\{z\} = \min\{10, 4 + \infty\} = 10$ $\therefore$ $L\{e\} = 5$ is the minimum label.

**Step 7 :** $v = e$, permanent label of $e$ is 5.

$P = \{a, b, f, c, d, e\}$, $T = \{g, z\}$

$L\{g\} = \min\{6, 5 + \infty\} = 6$ $\quad$ $L\{z\} = \min\{10, 5 + 3\} = 8$

$\therefore$ $L\{g\} = 6$ is the minimum label.

**Step 8 :** $v = g$, permanent label of $g$ is 6.

$P = \{a, b, f, c, d, e, g\}$, $T = \{z\}$

$L\{z\} = \min\{8, 6 + 1\} = 7$ which is the minimum label.

**Step 9 :** $v = z$, permanent label of $z$ is 7.

Hence the length of shortest path from a to z is 7.

The shortest path is a b f d g z.

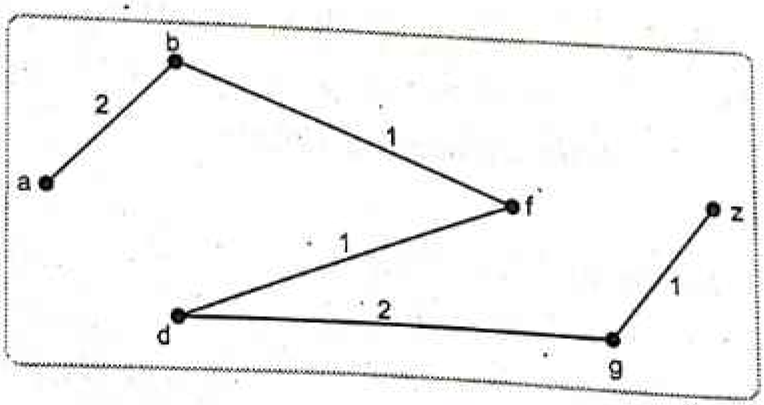



Fig. Q.32.2

**Q.33 Define Eulerian path and circuit.**

**Ans. :** A path is called an Eulerian path if every edge of graph G appears exactly once in the path.

A circuit of a graph which contains every edge of graph exactly once is called the Eulerian circuit.

A graph which has an Eulerian circuit is called as Eulerian graph.

The problem of find Eulerian path is the same as the problem of drawing a network without lifting the pencil off the paper and without retracing an edge.

Consider the following graphs :

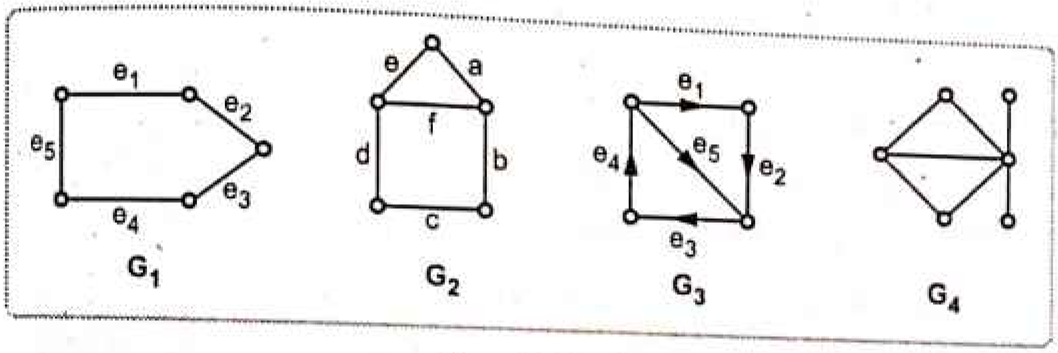



**Fig. Q.33.1**

In graph $G_1$, Eulerian circuit is $e - e_2 - e_3 - e_4 - e_5 - e_1$

$\therefore G_1$ is an Eulerian graph.

In graph $G_2$, Eulerian circuit does not exist.

$\therefore G_2$ is not Eulerian graph.

In graph $G_3$, Eulerian path is $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_4 \rightarrow e_5$ but $G_3$ does not have any Eulerian circuit.

$\therefore G_3$ is not Eulerian graph.

$G_4$ is also not an Eulerian graph.

The existence of Eulerian paths and circuits in a graph depends upon the degree of vertices.

**Theorem 1 :** An undirected graph possesses an Eulerian path iff it is connected and has either zero or two vertices of odd degree.

**Theorem 2 :** An undirected graph possesses an Eulerian circuit iff it is connected and its vertices are all of even degree.

**Theorem 3 :** A directed graph possesses an Eulerian circuit iff it is connected and incoming degree of every vertex is equal to its outgoing degree.

**Examples :**

**Q.34 a) Find under what conditions $K_{m,n}$ the complete bipartite graph will have an Eulerian circuit.**

**b) What is the complement of $K_{m,n}$ ?** [SPPU : Dec.-09]

**Ans. :** a) In $K_{m,n}$ consider the following cases.

**Case 1 :** m = n and both m and n are even :

In this case, degree of each vertex is even, Hence by theorem 1, $K_{m,n}$ will have an Eulerian circuit. For example $K_{1,2}$ and $K_{4,4}$

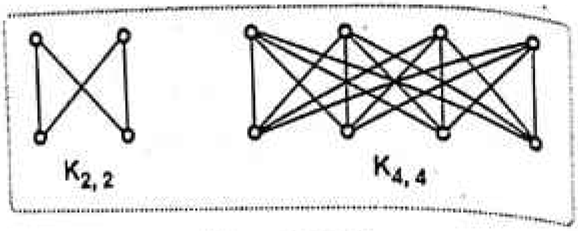



**Fig. Q.34.1**

**Case 2 :** If m = n and m, n are odd :

In this case degree of each vertex is odd. Hence Eulerian circuit will not exist.

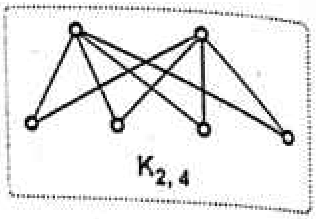



**Fig. Q.34.2**

**Case 3 :** If m ≠ n but m and n are even :

In this case, degree of each vertex is even. So there exists an Eulerian circuit.

**Case 4 :** If m ≠ n and either m is odd or n is odd or both are odd : Then graph will have vertices of odd degree. Hence Eulerian circuit does not exist. e.g. $K_2$, 3.

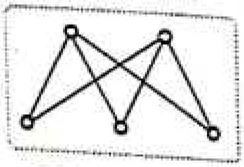


**Fig. Q.34.3**

b) The complement of $K_{m,n}$ is the two graphs $K_m$ and $K_n$.

**Q.35 Define Hamiltonian graphs.** [SPPU : Dec.-04, 09, 10, 12, 15]

**Ans. :** In this section, we introduce a class of graphs which possess a strinking similarity to Eulerian graphs.

We will now define Hamiltonian path and circuits of the connected graph.

A path in a connected graph G is called a Hamiltonian path if it contains every vertex of G exactly once.

A circuit in a connected graph G is called a Hamiltonian circuit if it contains every vertex of G exactly once.

A graph which has a Hamiltonian circuit is called a Hamiltonian Graph.

Consider the following graphs :

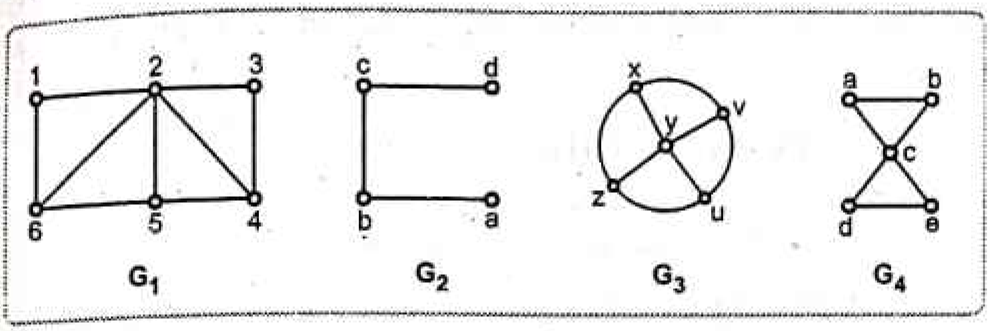



**Fig. Q.35.1**

In graph $G_1$, Hamiltonian circuit is 1-2-3-4-5-6-1

$\therefore G_1$ is a Hamiltonian graph.

In graph $G_2$, Hamiltonian path is a-b-c-d but Hamiltonian circuit does not exist.

$\therefore G_2$ is not Hamiltonian graph.

In graph $G_3$, Hamiltonian circuit is x-y-z-u-v-x

$\therefore G_3$ is Hamiltonian graph but it is not Eulerian.

In graph $G_4$, Hamiltonian path is a-b-c-d-e but Hamiltonian circuit does not exist.

$\therefore G_4$ is not Hamiltonian but Eulerian graph.

**Important theorems :**

**Theorem 1 :** Let G be a simple connected graph on n vertices. If the sum of the degree of each pair of vertices in G is $(n - 1)$ or large then there exists a Hamiltonian path in G.

**Theorem 2 :** If G (v, E) is a simple connected graph on n vertices and $d(v) = \frac{n}{2}$ ; $\forall v \in V$ then G will contain a Hamiltonian circuit.

This condition is sufficient condition but not necessary.

**Theorem 3 :** Let G (v, E) be a connected simple graph. If G has a Hamiltonian circuit then for every proper non empty subsets of v, the components in the graph G-S is less than or equal to the number of vertices in S.

**Theorem 4 :** A Hamiltonian graph contains no cut vertices and hence is 2-connected.

**Q.36 Show that the complete bipartite graph $K_{m,n}$ is Hamiltonian for m = n and for $m \neq n$, $K_{m,n}$ is not Hamiltonian graph.**

**Ans. :** In a complete bipartite graph $K_{m,n}$ for m = n i.e. $K_{n,n}$, degree of each vertex is n.

Therefore $d(v) \geq \frac{n}{2}$ for all $v \in v(K_{n,n})$

By theorem 2, G contains a Hamiltonian circuit.

Hence $K_{n,n}$ is a Hamiltonian graph.

If $m \neq n$, Let $V_1$ and $V_2$ be the partitions of the vertex set of $k_{m,n}$ where $|v_1| = m$ and $|v_2| = n$. Without loss of generality assume that $m < n$.

The graph $K_{m,n} - V_1$ is a null graph on n vertices.

Hence it is a disconnected graph with n components.

Therefore the number of components in $K_{m,n} - V_1 = n$ which is greater than the number of vertices in $V_1$

Hence by theorem 3, $K_{m,n}$ does not contain a Hamiltonian circuit when $m \neq n$.

**Q.37 Show that the complete graph $K_n$ $(n \geq 3)$ is a Hamiltonian graph. What is the length of that circuit ? How many circuits exist in $K_n$ ? What is the complement of $K_n$ ?** **[SPPU : Dec.-09]**

**Ans. :** The complete graph $K_n$ has n vertices, $n \geq 3$ and degree of each vertex is

$n - 1$. As $n \geq 3$.

$$d(v) = n - 1 \geq \frac{n}{2} ; \forall v \in V(K_n).$$

Therefore by theorem 2, $K_n$ has a Hamiltonian circuit. Hence $K_n$ is a Hamiltonian graph.

Hamiltonian circuit contains all vertices of graph and length of circuit is the number of vertices present it. Hence in $K_n$, the length of the Hamiltonian circuit is n and there are $\frac{(n-1)!}{2}$ Hamiltonian circuits in $K_n$.

The complement of $K_n$ is the null graph on n vertices.

**Q.38 Find the Hamiltonian path and circuit in $K_{4,3}$ ?**

**[SPPU : Dec.-04]**

**Ans. :** The complete bipartite graph $K_{4,3}$ is given by

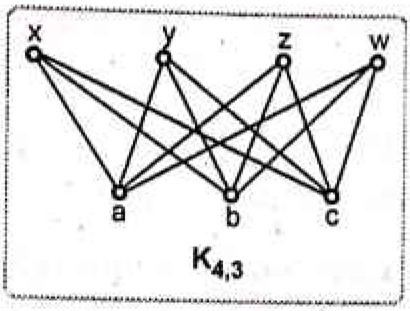



Fig. Q.38.1

In $K_{4,3}$, $4 \neq 3$ Hence it does not contain Hamiltonian circuit. Here degree of each vertex is either 3 or 4.

$\therefore$ For x, y any two vertices in $K_{4,3}$ $d(x) + d(y) = 7.1 = 6$

Hence by theorem 1, the graph $K_{4,3}$ has a Hamiltonian path. It is given by x a y b z c w.

**Q.39 Give an example of the following graphs**

**a) Eulerian but not Hamiltonian. b) Hamiltonian but not Eulerian. c) Eulerian as well as Hamiltonian. d) Neither Eulerian nor Hamiltonian.**

**Ans. : a) Eulerian but not Hamiltonian graph.**

Eulerian circuit : a b c d e c a

No Hamiltonian circuit.

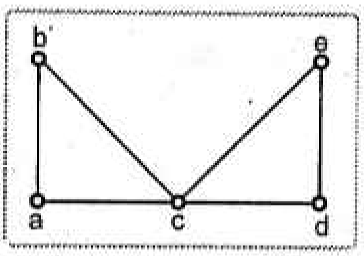



Fig. Q.39.1

**b) Hamiltonian but not Eulerian**

Hamiltonian circuit : abcdea

No Eulerian circuit because d (b) = 3.

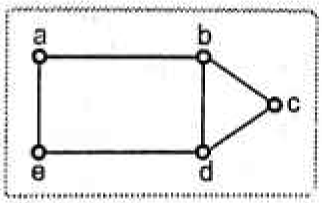



Fig. Q.39.1 (a)

**c) Eulerian and Hamiltonian graph.**

Hamiltonian circuit : a-b-c-a, 1-2-3-4-1

Eulerian circuit : a-b-c-a, 1-2-3-4-1

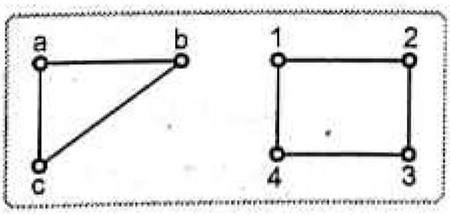



Fig.Q.39.1 (b)

d) Neither Eulerian nor Hamiltonian

Fig. Q.39.1 (c)

No Hamiltonian circuit and no Eulerian circuit.

**Q.40 Determine, if the following graphs are having the Hamiltonian circuit or path. Justify your answer.** [SPPU : Dec.-12]

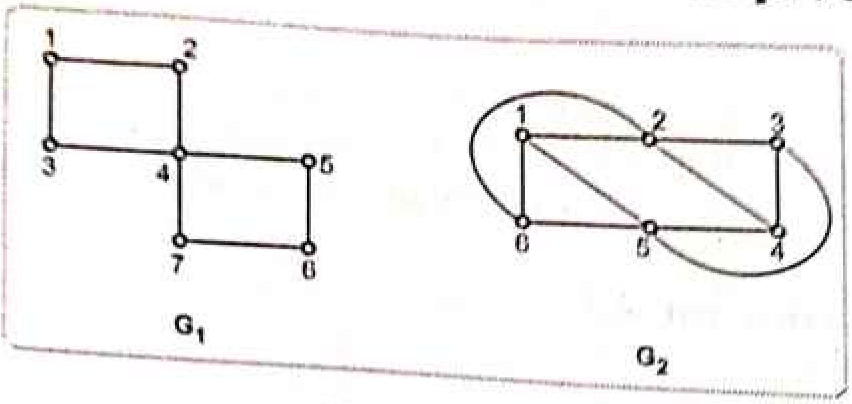



Fig. Q.40.1

**Ans. :** In graph $G_1$, there are n = 7 vertices.

d (4) = 4 and all remaining vertices is 2.

So to draw Hamiltonian circuit, we have to visit vertex 4 twice. Which is not possible in Hamiltonian path. $G_1$ has no Hamiltonian circuit. Hamiltonian path is 1-2-3-4-5-6-7.

In graph $G_2$, there are 6 vertices and degree of each vertex is 3 or 4.

If we consider two vertices of lowest degree then also their sum is 6 which is equal to the number of vertices. So there exists a Hamiltonian path in $G_2$. $\therefore$ Path is 1-2-3-4-5-6.

In graph $G_2$, $d(x) = \frac{6}{2} = 3$ ; $x \in V(G_2)$

$\therefore$ By theorem 2, $\exists$ a Hamiltonian circuit

$\therefore$ Hamiltonian circuit is 1-2-3-4-5-6-1.

**Q.41 Which of the following have a Euler circuit or path or Hamiltonian cycle ? Write the path or circuit.** [SPPU : Dec.-10]

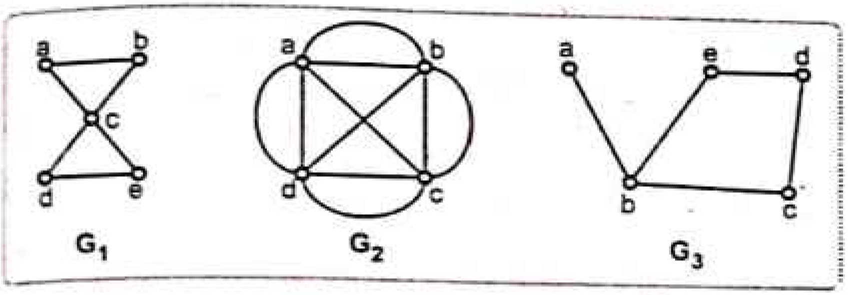



Fig. Q.41.1

**Ans. :** In graph $G_1$, degree of each vertex is an even so $\exists$ an Eulerian circuit which is a-b-c-d-e-c-a.

In graph G, there are 5 vertices and degree sum of every pair of vertices is 4 or greater than 4. Hence there exists a Hamiltonian path in $G_1$ which is given by a-b-c-d-e. But there is no any Hamiltonian circuit as vertex c is a vertex. In graph $G_2$, degree of each vertex is 5 which is odd integer, so there is no Eulerian path in $G_2$, degree of each vertex is $5 > \frac{4}{2}$. Hence there exists a Hamiltonian circuit which is given by a-b-c-d-a.

In $G_3$, Eulerian path is a-b-c-d-e-b No Eulerian circuit as d (a) = 1. Hamiltonian path is a-b-c-d-e.

No Hamiltonian cyle because b is a cut vertex.

## 3.3 : Travelling Salesman Problem (TSP)

**Q.42 Define the Travelling Salesman Problem (TSP).**

**Ans. :** A salesman is required to travel a number of cities during a trip. Given the distance among cities, in what order should he travel so that he travels as minimum distance as possible ? This is known as Travelling Salesman Problem (TSP).

In terms of graph theory, the TSP is to a Hamiltonian circuit with the smallest weight. In the case of $K_n$ the problem can be solved theoretically by listing all the possible Hamiltonian circuits and select one which has least weight. But this method is highly impractical for the large graphs. In fact no efficient algorithm is there to solve TSP. It is therefore desirable to obtain a reasonably good but not an optimal solution.

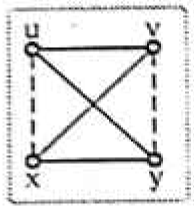



Fig. Q.42.1

One possible approach is to first find a Hamiltonian cycle and search for other Hamiltonian cycles of lesser weight. The simple method is as follows :

Let C be the Hamiltonian circuit of a graph G.

Let further uv and xy be two no-adjacent edges of C such that the vertices u, v, x and y occur in that order in C.

If ux and vy are edges such that v (u x) + w (y v) < w (uv) + w (yx) then replace the edges uv and xy in (by ux and vy). The new cycle C' would still be Hamiltonian cycle and w (C') < w (C). This process can be continued until one gets a reasonably good Hamiltonian cycle.

**Q.43 Explain nearest neighbour method.**

**Ans. :** In this method, we start with any arbitrary vertex and find the vertex which is nearest to it. Continuing this way and coming back to the starting vertex by travelling through all the vertices exactly once, we will get Hamiltonian cycle or circuit.

Consider the following steps to find Hamiltonian cycle by this method.

**Step 1 :** Start with any arbitrary vertex say $v_1$, choose the vertex closest to $v_1$ to form an initial path of one edge. Construct this path by selecting different vertices as described in step 2.

**Step 2 :** Let $v_n$ be the latest vertex that was added to the path. Select the vertex $v_{n+1}$ closest to $v_n$ from all vertices that are not in the path and add this vertex to the path. Select those vertices which will not form a circuit in this stage.

**Step 3 :** Repeat step (2) till all the vertices of G are included in the path.

**Step 4 :** Lastly form a circuit by adding the edge connecting to $v_1$ and the last added vertex.

The circuit obtained using the nearest neighbour method will be the required Hamiltonian circuit.

**Note :** If we start with an arbitrary vertex in TSP then we may or may not minimum Hamiltonian circuit, But if we start with a vertex whose incident edge has the minimum weight in graph then we will get minimum Hamiltonian circuit as compared with arbitary starting vertex. For more details see Q.44.

Examples :

**Q.44 Use nearest neighbour method to find the Hamiltonian circuit starting from a in the following graph. Find its weight.**

**[SPPU : Dec.-15]**

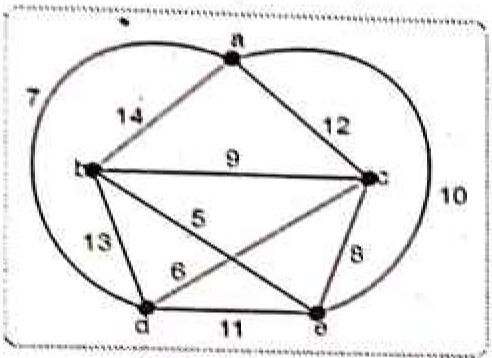



**Fig. Q.44.1**

Ans. : **Step 1 :** Let a be the starting vertex. Vertex a is adjacent to b, c, d, e. But minimum path is {a, d} which is the initial path.

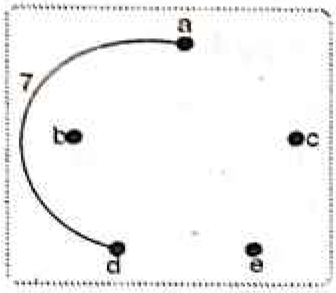



**Fig. Q.44.1 (a)**

**Step 2 :** There are three vertices adjacent to d. but closest one is C

∴ The path is {a, d, c}.

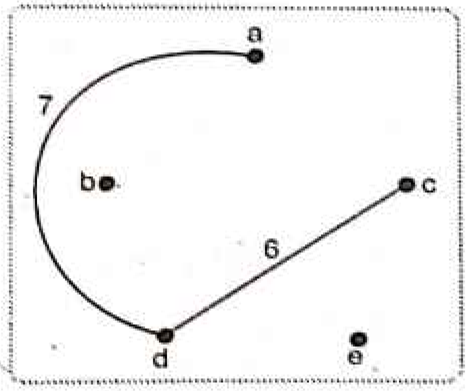



**Fig. Q.44.1 (b)**

**Step 3 :** There are 4 vertices adjacent to c. but closest is e

∴ The path is {a, d, c, e}.

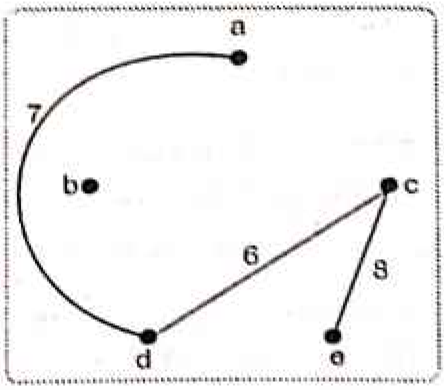



**Fig. Q.44.1 (c)**

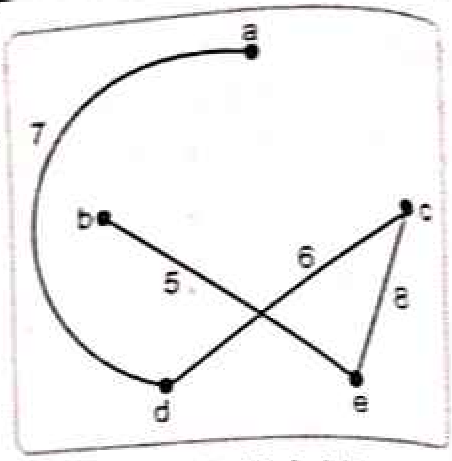



Fig. Q.44.1 (d)

**Step 4 :** There are 4 vertices adjacent to e. but closest is b

∴ The path is {a, d, c, e, b}.

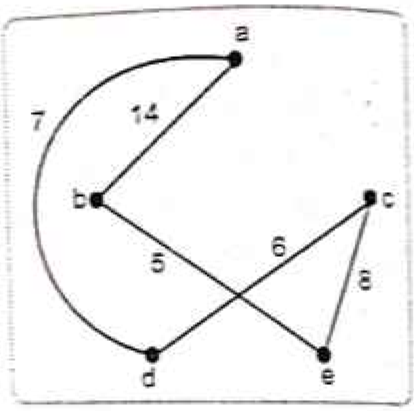



Fig. Q.44.1 (e)

**Step 5 :** Here are all vertices are corered so to complete Hamiltonian an circuit there should be a path from b to a.

∴ Hamiltonian circuit is {a, d, c, e, b, a}

Weight of the Hamiltonian circuit = 40.

## 3.4 : Planer Graph

**Q.45 Explain planar graphs.** ☞[SPPU : May-06, Dec.-08, 09, 10, 13]

**Ans. : Definition :** A graph is said to be planar graph if it can be drawn on a plane such that no edges intersect or cross in a point other than their end vertices.

A graph G is said to be non-planar if it is not possible to draw graph be without crossing.

**1. Regions :** A plane representation of a graph divides the plane into parts or regions. They are also known as faces or windows or meshes.
A region or face is characterised by the set edges forming it's boundary.

A region is said to be finite if it's area is finite. A region is said to be infinite or unbounded if it's area is infinite. Every planar graph has an infinite region.

Consider the graph given below :

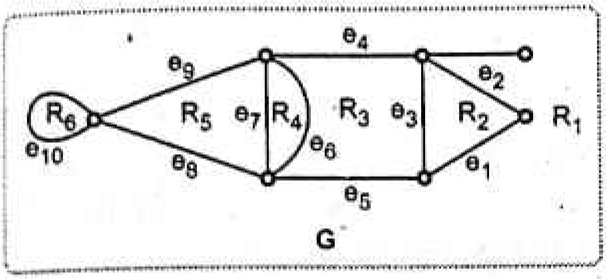



Fig. Q.45.1

The graph G has 6 regions, 7 vertices and 11 edges. Region $R_1$ is an infinite region known as exterior region. We have

$$R_2 = \{e_1, e_2, e_3\} = \text{Region bounded by } e_1, e_2, e_3$$

$$R_3 = \{e_3, e_4, e_5, e_6\}$$

$$R_4 = \{e_6, e_7\}, \quad R_6 = \{e_{10}\}.$$

It is observed that n = 7, e = 11, r = 6

$$\therefore \quad n + r - 2 = 7 + 6 - 2 = 11 = e$$

Now let us define Euler's formula.

**Q.46 State and prove Euler's formula.**

**Ans. : Statement :** For any connected planar graph G, with v number of vertices, e number of edges and r number of regions

$$\boxed{v - e + r = 2}$$

or $v + r - 2 = e$

**Proof :** Let G be a connected planar graph with v vertices, e edges and r regions. We shall prove the theorem by induction on e.

**Step 1 :** For e = 0, we get v = r = 1. Thus

$$v - e + r = 1 - 0 + 1 = 2$$

Hence result is true for e = 0

**Step 2 :** Let $e \geq 1$. Assume that the result is true for all connected planar graphs with less than e edges. Let G be a graph with v vertices, e edges and r regions.

**Step 3 : Case 1 :** If G has a pendent vertex say x then G – {x} is a connected graph with v – 1 vertices, e – 1 edges and r regions.

So by induction hypothesis

$$(v-1)-(e-1)+r = 2$$

$$v-e+r = 2$$

**Case 2 :** If G has no pendent vertex the G is a connected graph with circuit. Let $e_1$ be the edge of a circuit in G. Then G - $\{e_1\}$ is a connected graph with v vertices, e − 1 edges and r − 1 regions {If we remove edge from a circuit, then it reduces region by 1}.

By induction hypothesis

$$v-(e-1)+(r-1) = 2$$

$$\Rightarrow \quad v-e+r = 2$$

Thus by the principle of mathematical induction the result is true for all e.

**Q.47 If G (V, E) is a simple connected planar graph with v vertices and e edges then $e \leq 3v-6$.**

**Ans. : Proof :** Give that, G is a simple planar graph, so each region of G is bounded by three or more edges.

If G has r number of regions then the total number of edges in G is $e \geq 3r$.

Also each edge of G is included in exactly two regions of G. therefore $2e \geq 3r$

$$\Rightarrow \quad \frac{2e}{3} \geq r$$

Substitute these values in Euler's theorem, we get

$$v-e+r = 2$$

$$v-e+\frac{2e}{3} \geq 2$$

$$3v-e \geq 6$$

$$e \leq 3v-6$$ Hence the proof.

**Corollary 2 :** Prove that, $K_5$ (the complete graph on 5 vertices) is not planar.

**Proof :** The complete graph on 5 vertices $K_5$ is given below :

$K_5$ has 5 vertices and 10 edges. i.e. v = 5 and e = 10.

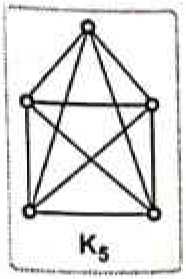



Fig. Q.47.1

Now $3v - 6 = 15 - 6 = 9$

By corollary 1, $e \leq 3v - 6$

$10 \leq 9$ which is impossible.

Therefore $K_5$ is not planar graph.

$K_5$ is the smallest planar graph with respect to number of vertices.

Consider the graph $K_{3,3}$

Here $v = 6, e = 9,$

$3v - 6 = 18 - 6 = 12 > 9 = e$

i.e. $e \leq 3v - 6$

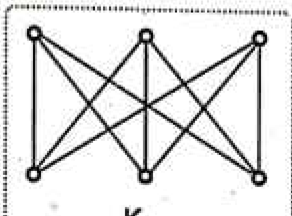



Fig. Q.47.2

But $K_{3,3}$ is not a planar graph.

$\therefore$ The graph $K_{3,3}$ is the smallest non planar graph with respect to number of edges.

The graph $K_5$ is called the Kuratowski's first graph and $K_{3,3}$ is called the Kuratowski's second graph.

In 1930, Kuratowski gave a necessary and sufficient condition for a graph to be planar.

**Kuratowski's Theorem :** A graph G is a planar if G does not contain any subgraph that is isomorphic to within vertices of degree two to either $K_5$ or $K_{3,3}$

Two graphs are said to be isomorphic to within vertices of degree two if they are isomorphic or they can be reduced to isomorphic graphs by repeated insertion of vertices of degree 2 or by merging the edges which have exactly one common vertex of degree 2.

For example the following graph are isomorphic to within vertices of degree 2. (Homeomorphic)

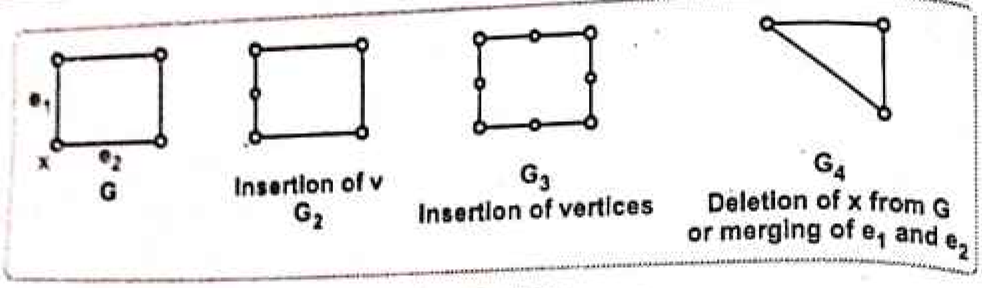



Fig. Q.47.3

**Q.48 Draw a planar representation of graphs given below if possible.**

[SPPU : May-06, Dec.-09]

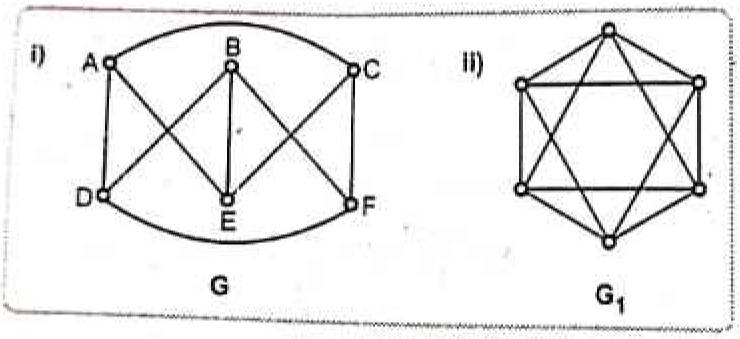



Fig. Q.48.1

**Ans. :** The planar representation of $G_1$ and $G_2$ is as follows :

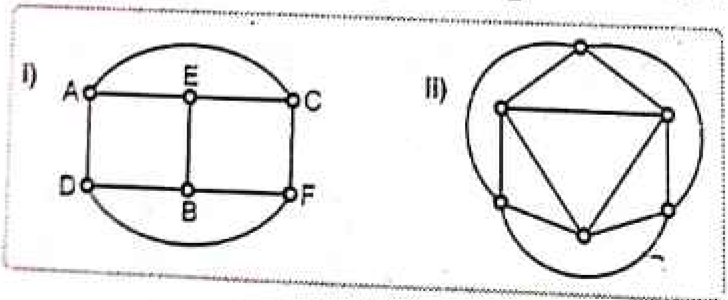



Fig. Q.48.1 (a)

**Q.49 Identify whether the graphs are planar or not Justify ?**

[SPPU : Dec.-08]

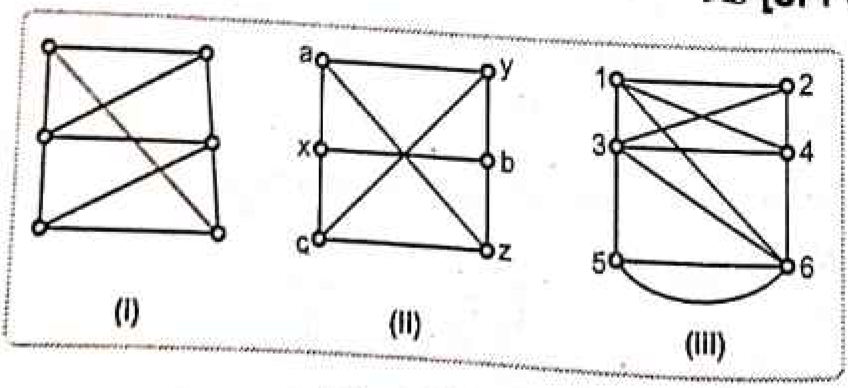



Fig. Q.49.1

**Ans. :** i)

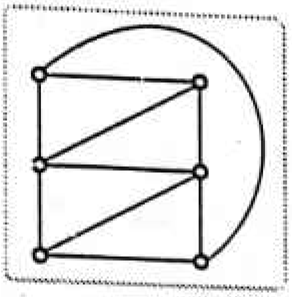


Fig. Q.49.1 (a)

Given graph is planar graph.

ii) Given graph is isomorphic to $K_{3,3}$

$\therefore$ Given graph is not planar.

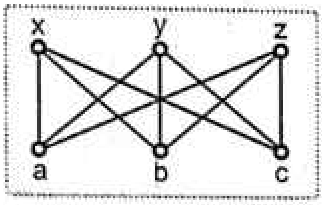



Fig. Q.49.1 (b)

iii)

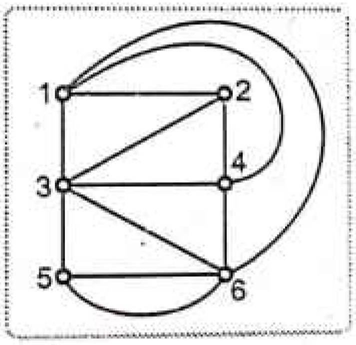



Fig. Q.49.1 (c)

$\therefore$Given graph is planar.

**Q.50 Show that in a connected planar graph with 6 vertices, 12 edges each of region is bounded by 3 edges.** **[SPPU : Dec.-10, 13]**

**Ans. :** According to Eulers theorem for planar graphs.

$$v - e + r = 2$$

Here
$$v = 6, e = 12$$

$6 - 12 + r = 2 \Rightarrow r = 8$

We know that, each edge contributed twice in a regions we have 12 edges.

So $12 \times 2 = 24$ edges are distributed among 8 regions.

$\Rightarrow \quad \frac{24}{8} = 3$ edges for each region.

So each region is bounded by 3 edges.

**Q.51 Prove that $K_{3,3}$ is not planar graph.**

**Ans. :** $K_{3,3}$ has 6 vertices and 9 edges. Suppose $K_{3,3}$ is planar, then the boundary of each region has at least 4 edges because it is bipartite and contains no triangles. Each edge lies on boundary of two regions.

Therefore, $2e \geq \sum_{i=1}^{r}$ (the number of edges in the $i^{th}$ region)

$$2e \geq 4r$$

$$2e \geq 4(2 + e - v)$$

$\Rightarrow \quad e \leq 2v - 4$ But $e = 9$ and $v = 6$

$\therefore \quad 9 \leq 12 - 4 = 8$ which is impossible.

Hence $K_{3,3}$ is not planar graph.

## 3.5 : Graph Colouring

**Q.52 Explain coloring of graphs.**

**Ans. :** The coloring of all vertices of a connected graph such that adjacent vertices have different colors is called a proper coloring or vertex coloring or simply a coloring of graphs.

A graph G is said to be properly colored graph if each vertex of G is colored according to a proper coloring.

e.g. 1) Consider the following graphs with proper coloring

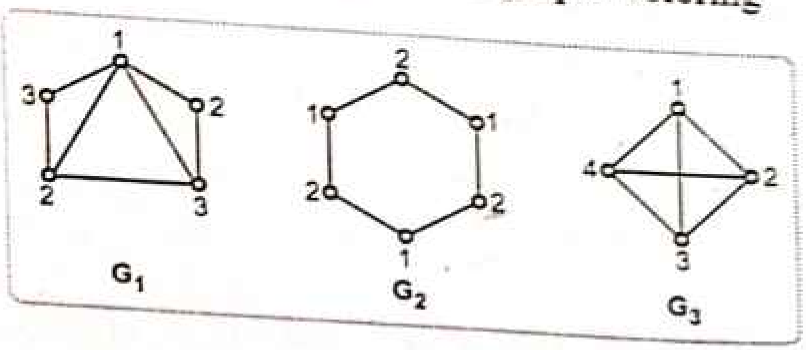



Fig. Q.52.1

**1. Chromatic Number of Graph :** The chromatic number of a graph G is denoted by X(G) and defined as the minimum number of colors required to color the vertices of G so that the adjacent vertices get different colors.

A graph G is said to be K-colorable if all vertices of G can be properly colored using at most K different colors. Obviously, a K-colorable graph is K+1 colorable.

If G is k-colorable then $X(G) \leq K$.

e.g. In above example (1) $X(G) = 3$, $X(G_2) = 2$, $G(G_3) = 4$. If G is any graph with $x(G) = K$ then the addition or deletion of loops or multiple edges do not change the chromatic number of that graph. Thus hereafter for a coloring of problem we consider only simple connected graphs.

**2. Chromatic Polynomial :** We have studied the properly coloring of graph in many different ways using a sufficiently large number of colors.

The chromatic polynomial of a graph is denoted by $P_n(\lambda)$ and defined as the number of ways of properly coloring of graph using $\lambda$ or fewer colors

e.g. 1) The chromatic polynomial of the complete graph $K_1$ is $P_1(\lambda) = \lambda$.

2) The chromatic polynomial of the complete graph $K_n$ on n vertices is

$$P_n(\lambda) = \lambda(\lambda - 1)(\lambda - 2) \ldots (\lambda - n + 1)$$

**3. Coloring of Planar Graph :** A map or atlas is a plane representation of a connected planar graph. Two regions of a planar graph G are said to be adjacent if they have an edge common.

The coloring of a planar graph or map means an assignment of a color to each region of a planar graph G such that adjacent regions have different colors.

A planar graph is - n colorable if minimum n different colors are required to color graph G.

**Theorem 1 :** (Four Color Theorem)

Every planar graph is 4 - colorable.

Initially it was a conjecture, but in 1979 Appel and Haken proved this. That's why this conjecture became theorem.

**4. Open Problem of Coloring :** A lot of research is done in the coloring of planar graphs, particularly coloring of vertices, or edges or regions of a planar graph.

The following open problem is stated by Dr. H. R. Bhapkar and proved partially first time.

**Open Problem :**

How many minimum colors will be required to color planar graph such that

i) Adjacent vertices have different colors

ii) Incident edges have different colors.

iii) Adjacent regions have different colors.

iv) A region, boundary edges and boundary vertices of that region have different colors.

This type of coloring is known as perfect coloring of G and denoted by PC (G).

We list some observations of perfect coloring of planar graph as follows :

i) If G is a null graph then PC (G) = 2

ii) If G is a chain graph when n vertices then,

$$PC(G) = \Delta(G) + 2$$

where $\Delta(G)$ = Highest degree of a vertex in G.

**Q.53 Is each of the following graphs strongly connected.**

[SPPU : Dec.-16]

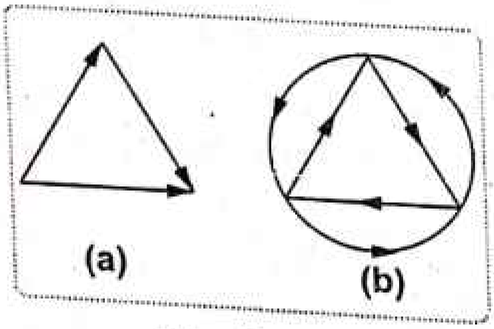


Fig. Q.53.1

**Ans. :** a) Graph is not strongly connected.

b) Graph is strongly connected.

## 3.6 : Tree

In this chapter we will study one of the simplest types of the connected graphs known as trees. This class of graphs has wide applications and has been the subject of study of many outstanding scientiests of different fields. Trees are discovered by Kircnnoff in 1847, while investigating the electrical networks. Sir Arthur Cayley used trees to study and enumerate isomers of saturated hydrocarbons.

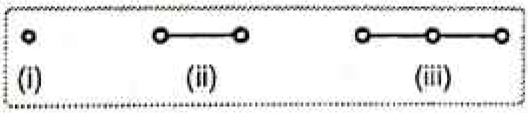
The important applications of tree include searching, sorting, syntax checking and database managements. The tree is one of the most non-linear structures used for algorithm development in computer science.

**Q.54 Define tree with examples.**

**Ans. :**

A tree is a connected graph without any circuit i.e. tree is a connected acyclic graph. The collection or set of an acyclic graphs. (not necessarily connected) is called a forest.

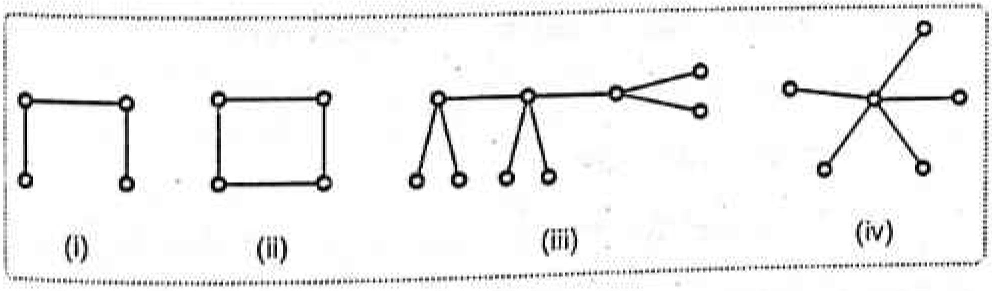
**Examples :**




**Example 1**

All these three graphs are trees. There are unique tree on one vertex, 2 vertices and 3 vertices.

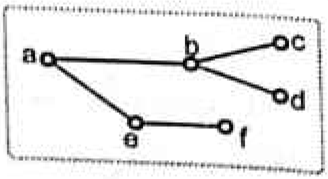
**Example 2**




Graphs (i), (iii) and (iv) are trees but (ii) is not a tree as it has a cycle.

A) A vertex of degree 1 in a tree is called a **leaf** or a **terminal node**. A vertex of a degree greater than one is called a **branch node** or **internal node.**

e.g. In a tree




c, d and e are leaves or terminal nodes and a, b, c are branch nodes.

B) Some properties of tree are obvious

i) Every edge of a tree is an isthmus. Conversely if every edge in a connected graph G is an isthmus then G is a tree.

ii) Every tree is a simple graph because loops and multiple edges field cycles.

iii) Every tree is a bipartite graph with $n \geq 2$ vertices..

**Theorem 1 :** A graph G is a tree iff there exists a unique path between every distinct pair of vertices of G.

**Proof :** Suppose G is a tree. So G is a connected and without circuits. We know that a circuit forms two or more paths. But G has no circuit so it has unique path between every pair of vertices.

Conversely assume that there is a unique path between every pair of vertices in G. This implies that G is a connected and G has no cycles. i.e. G is a connected acyclic graph.

Therefore G is a tree.

**Theorem 2 :** A graph G on n vertices is a tree iff G is connected and has exactly $n - 1$ edges.

**Proof :** Suppose graph G is a tree. Therefore is a connected and acyclic graph.

It is sufficient to prove that G has $n - 1$ edges only.

We can prove this by induction principle on n.

For $n = 1$, the result is obvious

Let $n > 1$, consider $G - e'$ for any $e' \in E(G)$

As G is a tree, $e'$ is an isthmus of G.

$\therefore G - e'$ is a disconnected graph with two components say $G_1$ and $G_2$.

Now $G_1$ and $G_2$ are connected and a cyclic graphs as $G_1$ and $G_2$ are subgraph of G.

Let $G_1$ has $n_1$ vertices and $m_1$ edges and $G_2$ has $n_2$ vertices and $m_2$ edges

$\therefore$ By induction principle

$$m_1 = n_1 - 1 \qquad \text{and} \qquad m_2 = n_2 - 1$$

Therefore the number of edges in G is given by

$$e = (e_1 + e_2) + 1 = n_1 - 1 + n_2 - 1 + 1$$
$$= n_1 + n_2 - 1$$
$$e = n - 1 \qquad (\because n = n_1 + n_2)$$

Hence G has n – 1 edges only.

Conversely assume that G is connected graph and has exactly n – 1 edges.

Claim : Prove that G is a tree.

It is suffices to prove that G is noncyclic graph.

Suppose G contains a cycle C.

Let P denote the number of vertices in C

$\therefore$ The number of edges in C = P

As G is connected graph, the remaining n – P vertices must be connected to vertices in C. To connect each vertex of G which is not in C, we required n – P edges as each edge of G can connect only one vertex to the vertices in C.

Hence the total number of edges in G is given by

$$e = (n - P) + P = n \Rightarrow e = n \text{ which is contradiction}$$

$\therefore$ G has n – 1 edges only.

Thus G is a tree.

**Theorem 3 :** Let G be a graph with n vertices and m edges. If any of the following is true then all are true.

i) G is a tree.

ii) G is connected and m = n – 1

iii) G is acyclic graph and m = n – 1

iv) Every edge of G is an isthmus and G is connected.

v) There is exactly one path between every pair of vertices in G.

**Theorem 4 :** A non trivial tree contains at least two vertices of degree 1. (i.e. pendent vertices)

**Proof :** Let G be a tree with n vertices, so G is connected.

$\therefore \quad d(x) \geq 1 \ ; \ \forall x \in V\ (G)$

By handshaking lemma

$$\sum_{x \in V(G)} d(x) = 2 \times \text{Number of edges in } G = 2(n-1)$$

Suppose there is no vertex of degree 1.

Then $$2n \leq \sum_{x \in V(G)} d(x) = 2n - 2$$

i.e. $2n \leq 2n - 2 \Rightarrow n \leq n - 1$ which is impossible.

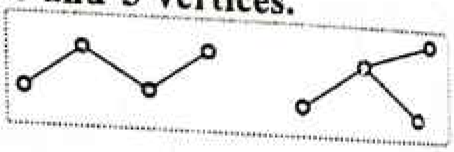
Thus there is at least one vertex of degree 1.

By a similar argument there is one more vertex of degree 1.

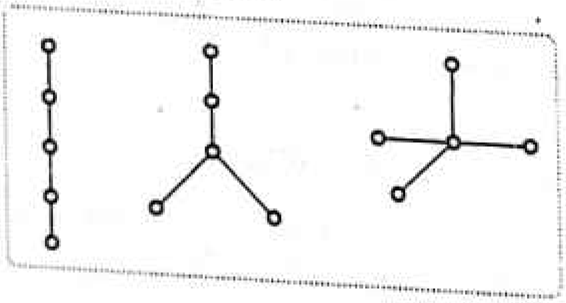
Hence every tree has at least two pendent vertices.

**Q.55 Draw all non isomorphic trees on 4 and 5 vertices.**

**Ans. :** a) Non isomorphic trees on 4 vertices



b) Non isomorphic trees on 5 vertices



**Q.56 Under what conditions trees are the complete bipartite graphs.**

**Ans. :** Suppose T is a tree which is the complete bipartite graph.

Let $$T = K_{m,\,n}$$

$\therefore$ T has $m + n$ vertices and $(m+n-1)$ edges.

But $K_{m,\,n}$ has $m \cdot n$ number of edges.

Therefore
$$mn = m + n - 1$$
$$mn - m - n + 1 = 0$$
$$(m-1)(n-1) = 0$$

$$m = 1 \text{ or } n = 1$$

$\Rightarrow$

i.e. $\quad m = 1$ and any $n \quad$ or $\quad n = 1$ and any $m$

Hence $K_{1,n}$ and $K_{m,1}$ are the only complete bipartite graphs. These are known as star graphs.

Thus T is a star graph.

**Q.57 a) Is it possible to draw a tree with 10 vertices which has vertices either of degree 1 or 3 ?**

**If possible draw tree. Is it possible to draw same type of tree with 13 vertices ?**

**b) For which values of n (number of vertices), such type of tree exist ?**

**Ans. :** **a)** Given that tree T has 10 vertices so it must have 9 edges.

Let x and y be the number of vertices of degree 1 and 3 in T respectively.

$$\therefore \quad x + y = 10 \qquad \text{... (Q.57.1)}$$

By handshaking lemma

$$\sum_{V \in V(T)} d(v) = 2(\text{Number of edges in T})$$

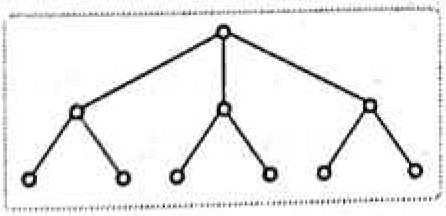
$$x + 3y = 2 \times 9 = 18$$

$$x + 3y = 18 \qquad \text{... (Q.57.2)}$$

Solving equations (Q.57.1) and (Q.57.2) we get

$$y = 4 \quad \text{and} \quad x = 6$$

Therefore there are 6 vertices of degree 1 and 4 vertices of degree 3. Such type of graph is given below.



Now consider a tree with 13 vertices and 12 edges.

By using similar theory, we get

$$x + y = 13 \qquad \text{... (Q.57.3)}$$

$$x + 3y = 24 \qquad \text{... (Q.57.4)}$$

Solving (Q.57.3) and (Q.57.4), we get $1y = 11 \Rightarrow y = \frac{11}{2}$ and $x = \frac{15}{2}$ which is impossible.

Therefore it is not possible to draw such type of tree.

b) Let T be a tree with n vertices.

Let x and y be the number of vertices of degree 1 and 3 in T respectively. T has $n - 1$ edges.

Therefore $\quad x + y = n \quad$ ... (Q.57.5)

By handshaking lemma

$$x + 3y = 2(n - 1)$$

$$x + 3y = 2n - 2 \quad \text{... (Q.57.6)}$$

Equation (Q.57.6) – Equation (Q.57.5) $\Rightarrow 2y = 2n - 2 - n$

$$2y = n - 2$$

$$\Rightarrow \quad y = \frac{n-2}{2} = \frac{n}{2} - 1$$

$$\Rightarrow \quad x = n - y = n - \frac{n}{2} + 1 = \frac{n}{2} + 1$$

i.e. $x = \frac{n}{2} + 1$ and $y = \frac{n}{2} - 1$ These should be non negative integers.

**Case 1 :** If n is even then $x = \frac{n}{2}$ is odd

and $\quad \frac{n}{2} + 1 = x$ is an even integer

$\frac{n}{2} - 1 = y$ is an even integer

Thus the tree exists with required condition on even number of vertices. $T_2, T_4, T_6, T_8 \ldots T_{2n}$ are such type of trees.

**Case 2 :** If n is odd then $\frac{n}{2}$ is not an integer there x and y are not an integer.

Hence the required tree does not exist on odd number of vertices.

**Q.58 Explain centre of a tree.**

**Ans. :** We know that the distance between two vertices in a connected graph i.e. is a length of the shortest path between that vertices. Before defining centre of a tree first find the distance between vertices of a tree which is defined as follows :

### Eccentricity of a Vertex

Let G be a connected graph G and $v \in V(G)$. The eccentricity of a vertex v is denoted by E(v) or e(v) and defined as the distance from v to the vertex farthest from v in G.

i.e. $E(v)$ or $e(v) = \max\{d(v, v_i)/\forall v_i \in v(G)\}$

### Centre of a Graph

A vertex in a graph G with minimum eccentricity is called a centre of G and it's eccentricity is called as radius of G. It is denoted by r(G).

**Q.59 Prove that every tree has either one or two centres.**

**Ans. :**

**Proof :** Let T be a tree. Then e(v) for a vertex v is at a vertex farthest from v. As T is a tree, it is attained at a pendent vertex.

We know that every non trivial tree has at least two pendent vertices. Now, delete all pendent vertices from G, we get new graph $G'$ which is also a tree. Moreover as one edge at each pendent vertex is removed, the eccentricity of any vertex will be reduced by one. Therefore centres of G will still remain centres of $G'$. We continue the process with $G'$ until we arrive at $K_2$ or $K_1$. $K_2$ has two centres and $K_1$ has one centre. Hence the proof.

**Q.60 Define cut vertex of a tree.**

**Ans. :** We know that the vertex v whose removal from a connected graph G, disconnects the graph is called as cut vertex of G.

In any tree all vertices except pendent vertices are cut vertices.

## 3.7 : Rooted Tree

**Q.61 Define rooted tree and binary tree.** [SPPU : Dec.-09]

**Ans. :** A connected acyclic, directed graph is called a **directed tree.** In other words, a directed graph is said to be a directed tree if it will become a tree when the directions of the edges are ignored.

**Q.62 Define rooted tree.**

**Ans. :** A directed tree is called a **rooted tree** if there is exactly one vertex whose incoming degree is zero and the incoming degrees of all other vertices are one.

The vertex with incoming degree 0 is called **the root** of the rooted tree. The vertex whose outgoing degree is zero is called **leaf or terminal node.**

A vertex whose outgoing and incoming degrees are non zero is called a **branch note** or an **internal node.** Consider the following example.

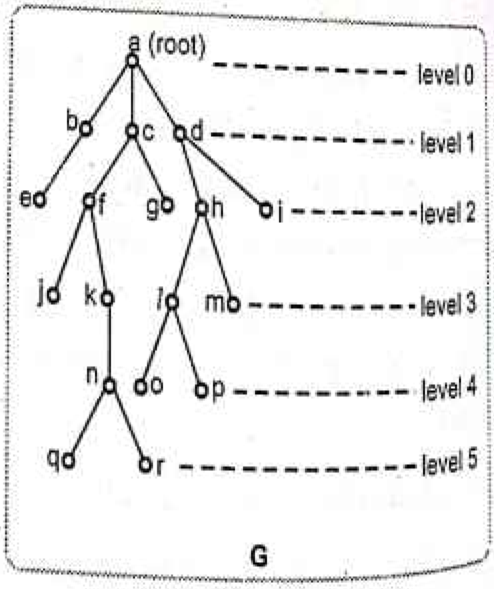
**Q.63 Define level and height of a tree.**

**Ans. :** A vertex v in a rooted tree is said to be at **level n** if there is a path of length n from the root to the vertex v.

The **height** of the tree is the maximum of the levels of its vertices.

**Example**

In the given graph G .... a is a root




Vertices b, c, d are at level 1

Vertices e, f, g, h and i are at level 2

Vertices j, k, l and m are at level 3

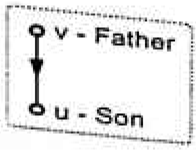
Vertices n, o and p are at level 4

Vertices q and r are at level 5

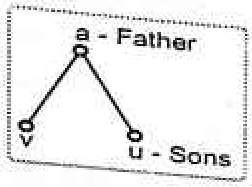
The maximum level is 5. Therefore the height of tree is 5.

Rules for rooted tree : In a rooted tree

i) If the level of a vertex u is greater than the level of vertex v then u is below v.




ii) If vertex u is below vertex v and there is an edge from v to u then u is said to be son of v (or child of v) and v is said to be the father of u (or parent of u) i.e.

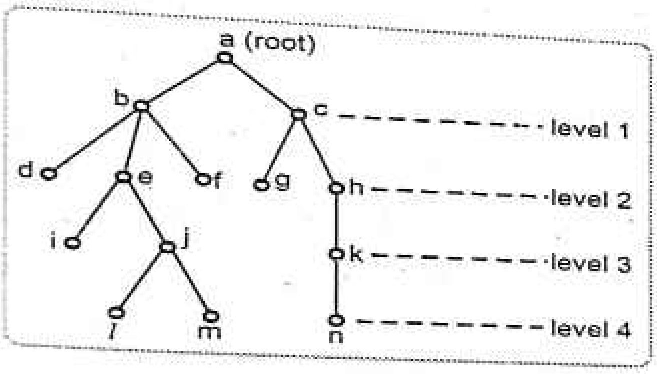



iii) Two vertices u and v are said to be brothers if they are the sons of the same vertex.

iv) A leaf is a vertex without children.

v) If p = {a, $v_1$, $v_2$, $v_3$, .... $v_{n-1}$, b} is a path from a to b then b is called as descendent of a and a is called as ancestor of b.

Consider the following example




From above figure i) a is a root of the tree.

ii) b and c lie at level 1. ∴ b and c are sons of root a i.e. a is father of b and c and b and c are brothers.

iii) b has three sons d, e, f

∴b is a father of d, e, f

c has 2 sons g and h

iv) e has two sons i and j

h has only one son k

v) i has no son j has two sons $l$ and m. So k and $l$ are brothers. k has one son n. n has no brother as n is a leaf.

$l$, m, n are known as descendent of a and a is ancestor of d, e, f, g, h, i, j, k, $l$, m, n and so on.

**Q.64 Define Subtrees.**

**Ans. :** Let T be a rooted tree and $x \in V(T)$.

A vertex x together will all its descendants is called the **subtree of T** rooted at x.

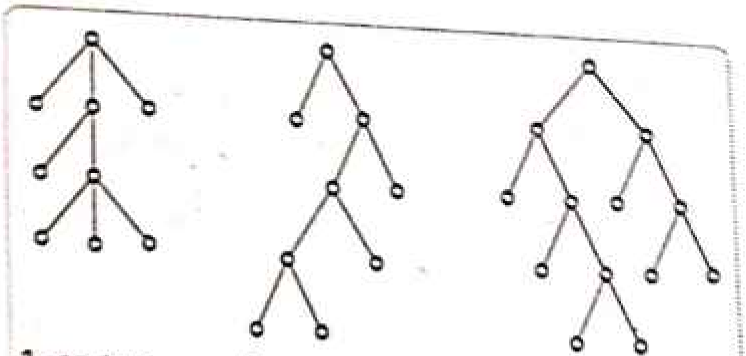
The subtree corresponding to the root node is the **entire tree.** The subtree corresponding to any other node is called a **proper subtree.**

**Q.65 Define M-ary trees.**

**Ans. :** A rooted tree in which every interior node has at most m sons is called an **m-ary tree.**

A m-ary tree is said to be **regular m-ary tree** or full m-ary tree if every branch node has exactly m sons.

Consider the following examples.



3 - ary tree 2 - ary tree Full 2 - ary tree

**Q.66 A regular m-ary tree with p interior nodes has mp + 1 nodes at all.**

[SPPU : Dec.-09]

**Ans. :**

**Proof :** Let T be a regular m-ary tree with n vertices. Out of n vertices there are p interior vertices or branch nodes.

Therefore there are $t = n - p$ number of sons or leaves in T.

But given graph is regular and p interior nodes.

So the regular m-ary tree will have mp sons.

But root is not a son.

Therefore give tree has total (mp + 1) number of vertices

Hence

$$\boxed{n = mp + 1}$$

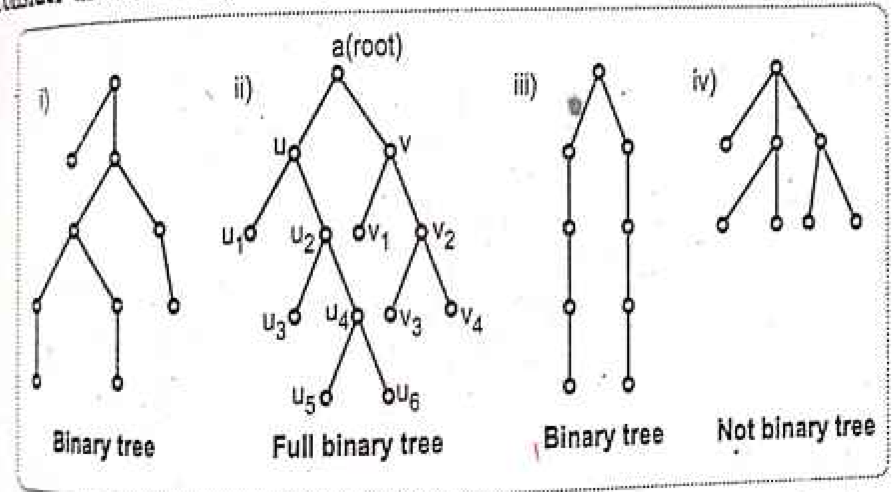
**Q.67 Explain binary tree.** [SPPU : Dec.-09, May-10, 12]

**Ans. :** An m-ary tree is known as **binary tree** if every branch node has at most 2 sons.

In other words, a tree in which there is exactly one vertex of degree 2 and each of the remaining vertices of degree or three, is called a binary tree.

A binary tree is called as **regular binary tree** or **full binary tree** if every branch node has exactly 2 sons or zero son.

Consider the following examples



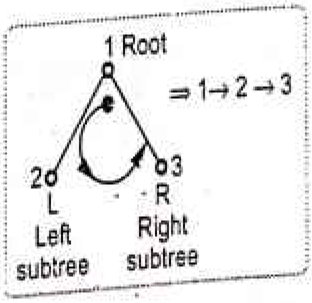

In binary trees, instead of referring the first or second subtree of a branch node, we use to the left subtree or right subtree of the node.

**Q.68 Explain binary tree traversal with examples.**

**Ans. :** Traversing means visiting or processing all the nodes of a tree. A binary tree traversal is the visiting of each node of a tree only once according to some sequence.

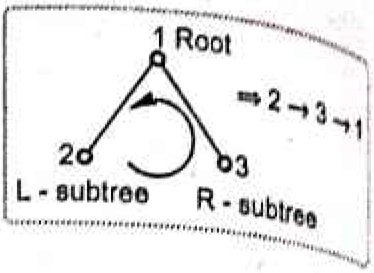
There are two types of traversing binary trees.

**1) Depth-first traversal :** In this method, the processing proceeds from the root or the most distant descendent of the first child.

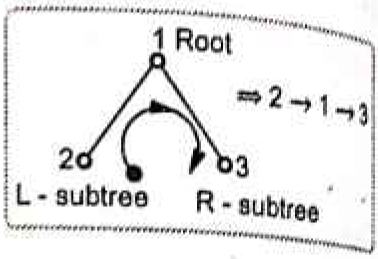


There are three types of Depth-First Traversal.

**A) Pre order traversal :** In this traversal, the root node is traversal first, followed by the left subtree and then the right subtree as shown below.



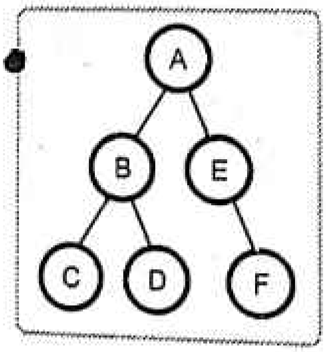

**B) Post order traversal :** If processes first the left subtree then the right subtree and then at the last root of the tree as shown below.



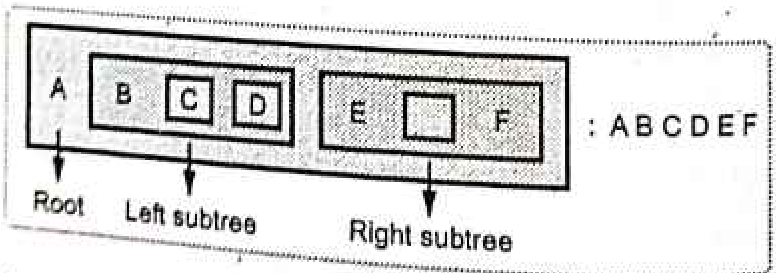

**C) In order traversal :** It processes the left subtree first then the root and at the last right subtree.

The prefix "in" means root is processed in between the subtrees. It is shown as below.

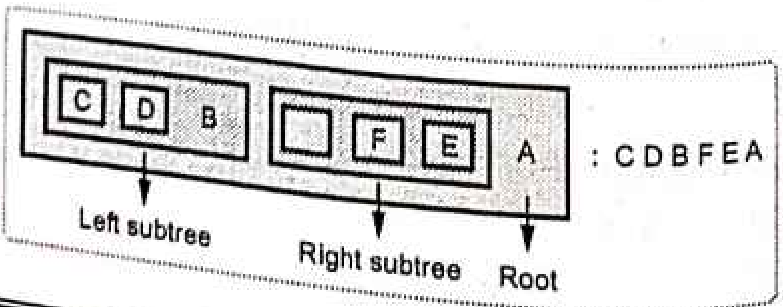
Consider the following binary tree.



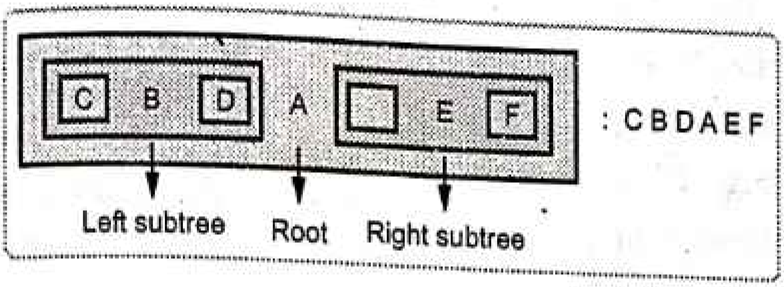

i) Pre order traversal



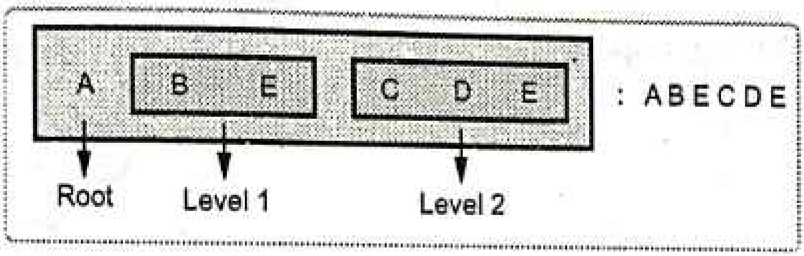

ii) Post order traversal



iii) In order traversal

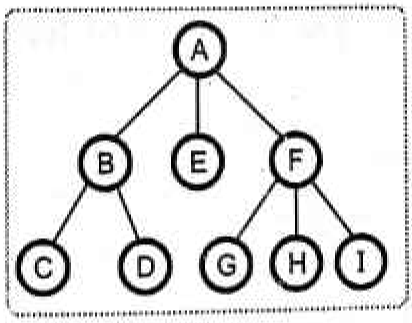



**2) Breadth first traversal :** In this traversal, the processing proceeds horizontally from the root of all of its children, then to its childre's children and so on until all the nodes have been processed. That is first write node of zero level, then all nodes of level 1 then level 2 and so on. The breadth first process for the above binary is shown below.



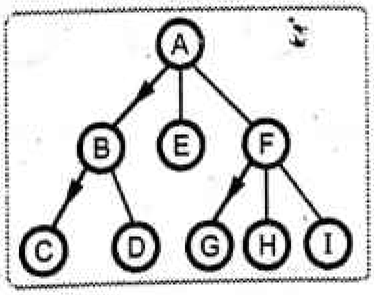

**Q.69 Explain conversion of general tree to binary tree.**

**Ans. :** Let us explain the method to convert general tree to binary tree with the help of the following example.

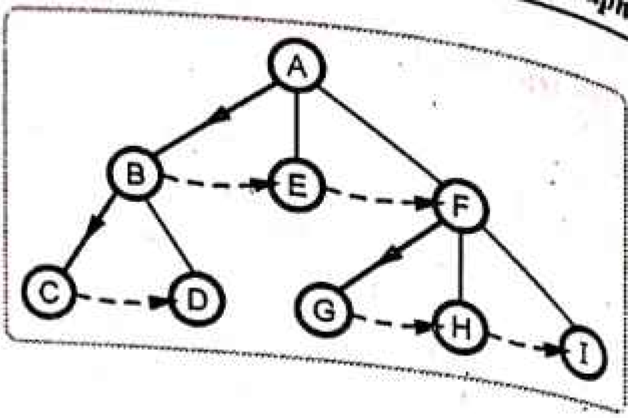



Consider the following steps

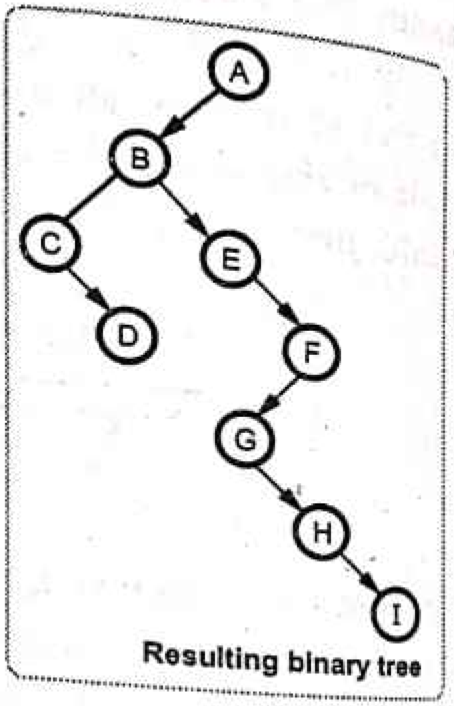
Step 1) To convert it into binary tree, we first identify the branch from the parent to its first or left most child. These branches from each parent become left pointers in the binary tree.



Step 2) Connect sibling, starting with the left most or first child, using a branch for each sibling to its right sibling. They are the right pointers in the binary tree.

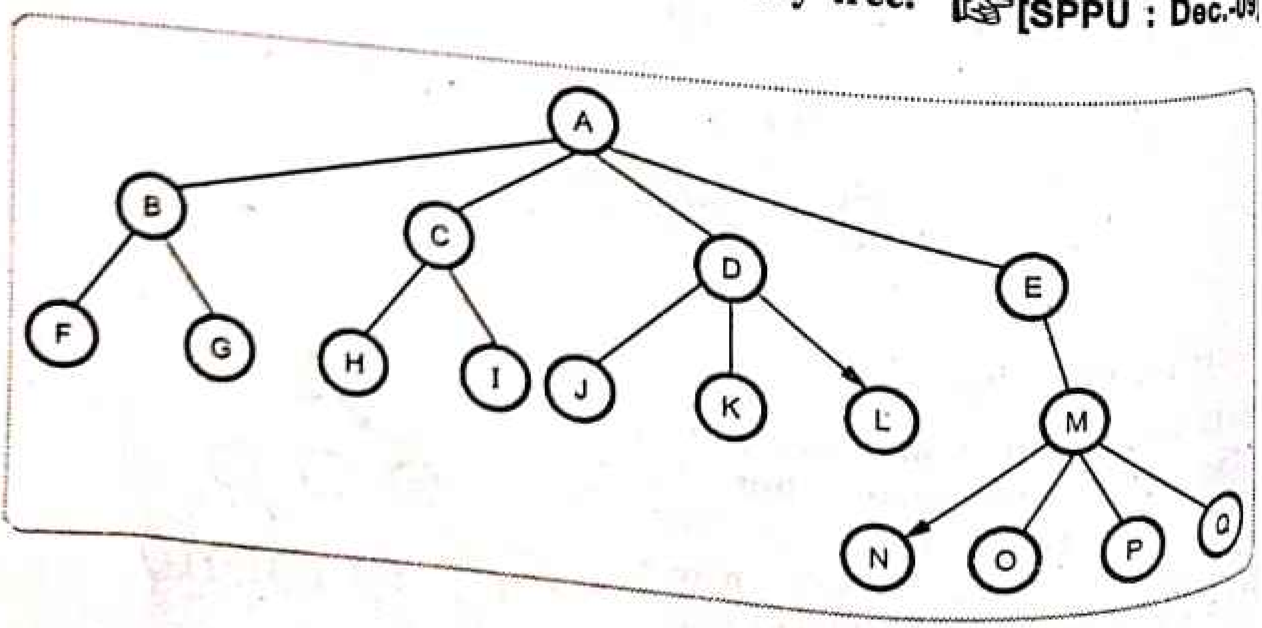



Step 3) Now remove all unneeded branches from the parent to its children. Therefore remove $A \rightarrow E$, $B \rightarrow D$, $A \rightarrow F$, $F \rightarrow H$, $F \rightarrow I$ we get the following required binary tree.

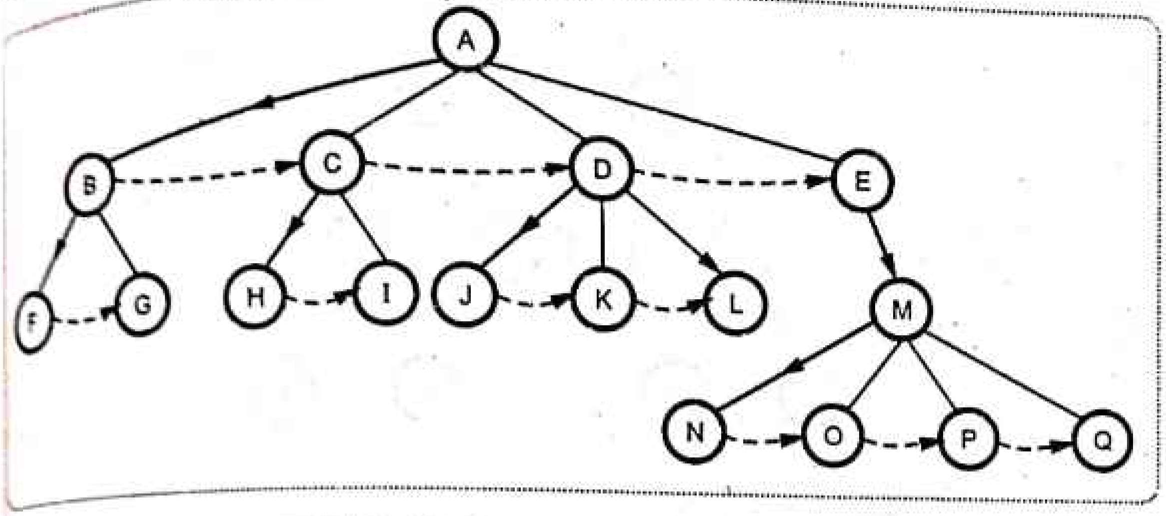


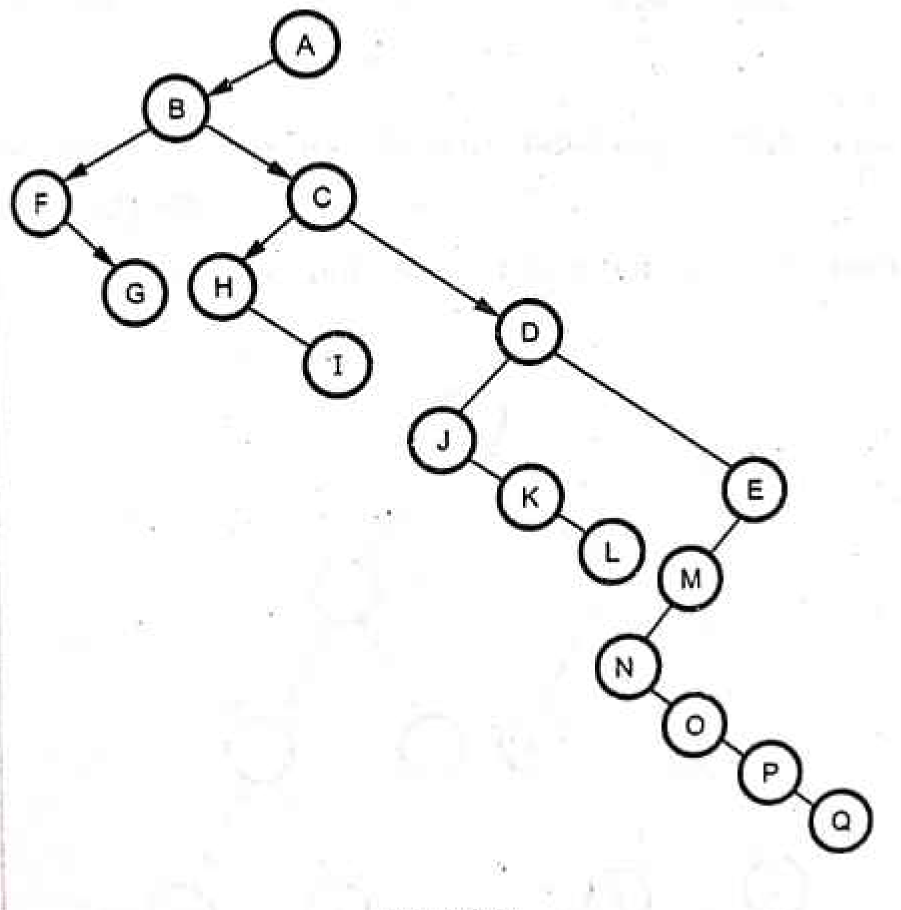

Resulting binary tree

**Q.70 Convert the following tree into binary tree.** ☞[SPPU : Dec.-09]



Ans. : The steps involve to convert the given tree into binary tree are as follows

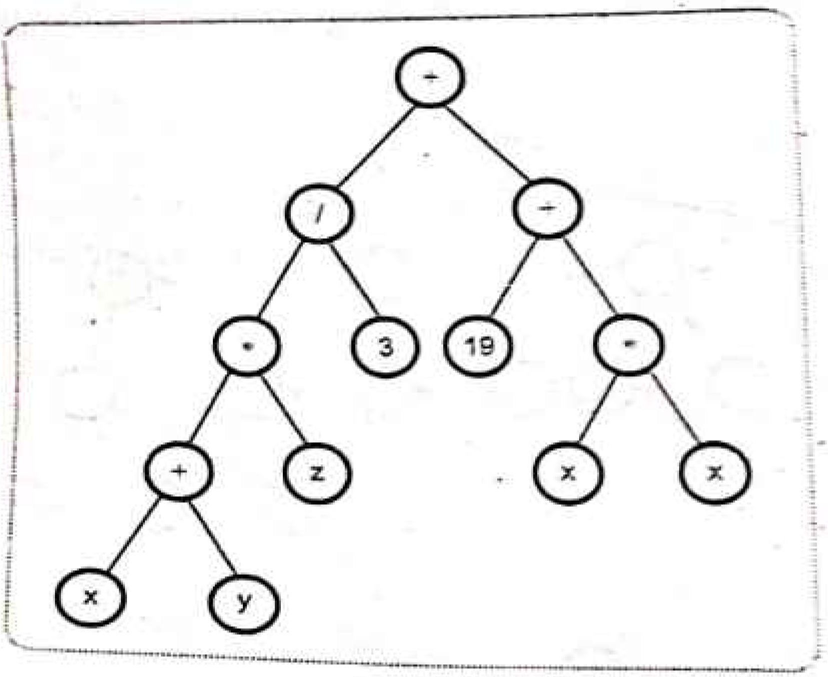



Binary tree

**Q.71 Construct the labeled tree of the following algebraic expression $(((x+y)*z)/3)+(19+(x*x))$** [SPPU : May-10]

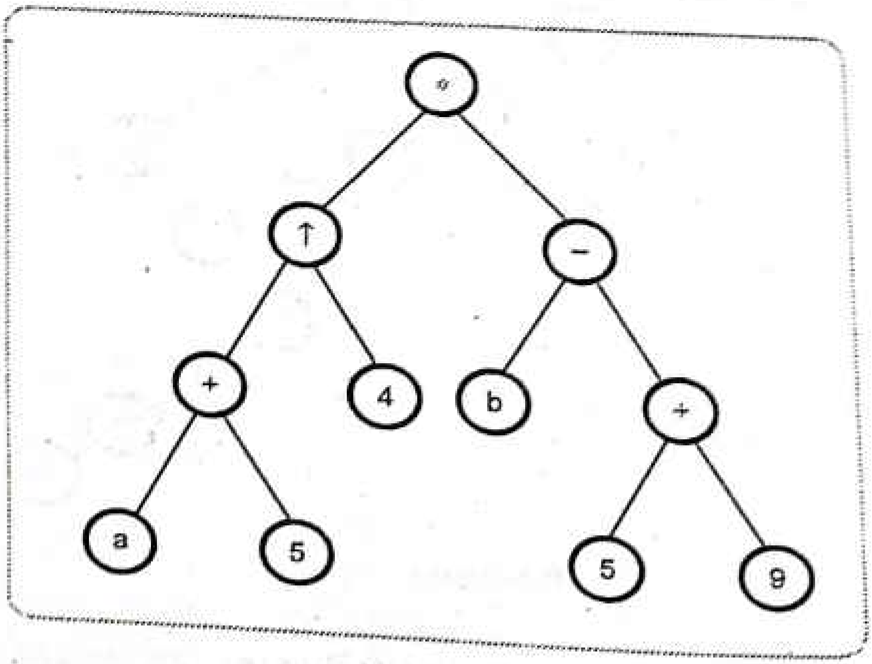
Ans. : The expression tree for the given algebraic expression is given below :



**Q.72 Represent the expression $((a+5)\uparrow 4)*(b-(5+9))$ using a binary tree.**

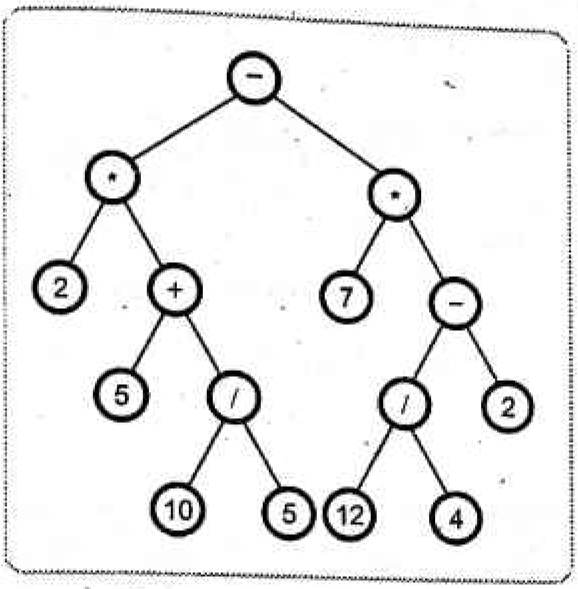
[SPPU : May-12]

**Ans. :** The binary tree for the given expression is as follows :



**Q.73 Write and evaluate the expression tree shown below.**

[SPPU : Dec.-09]

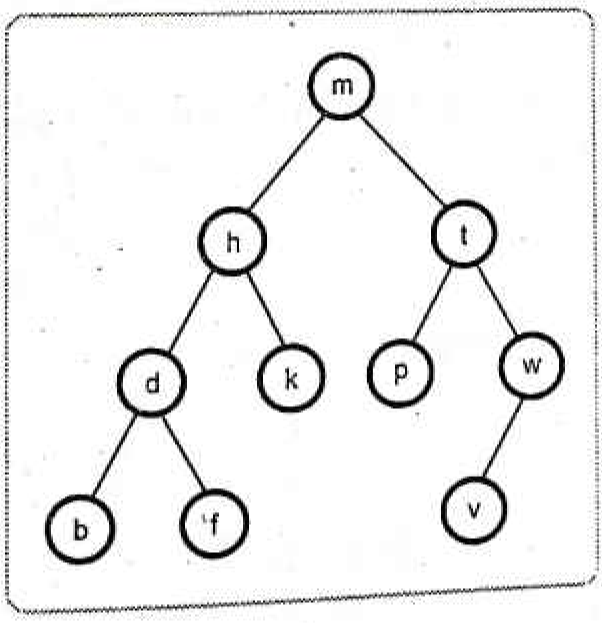



**Ans. :** The algebraic expression of the given binary tree is

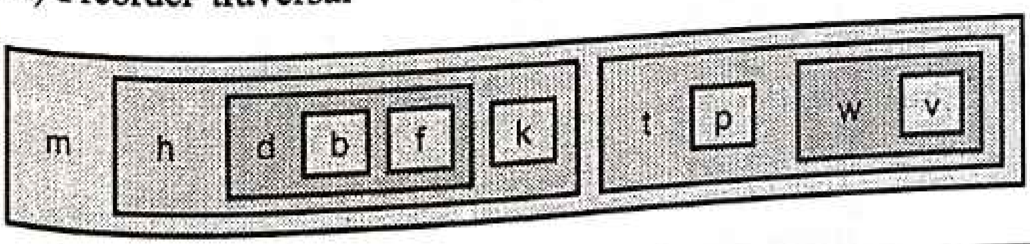
$$(2*((5)+(10/5)))-((7)*(((12/4)-(2))))$$

It's value is $2*(5+2)-(7*(3-1)) = 14 - 7 = \mathbf{14}$

**Q.74 Find the preorder, postorder and inorder traversal of the following tree.** **[SPPU : May-10]**

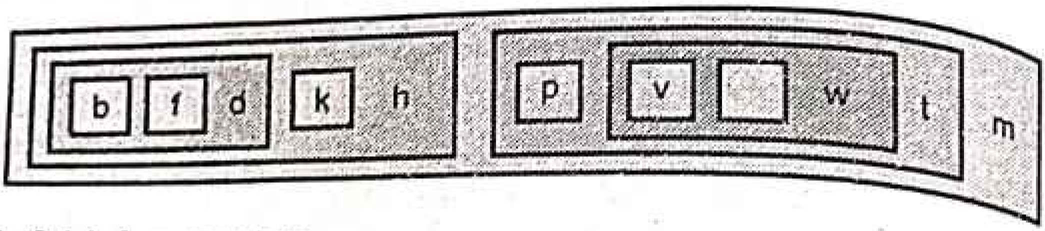



**Ans. :** i) Preorder traversal



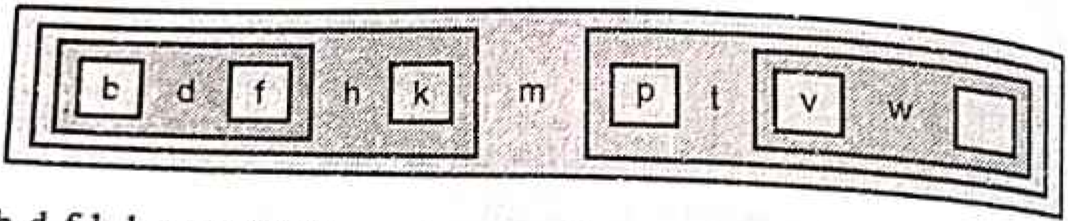
i.e. m b d b f k t p w v

ii) Postorder traversal




i.e. b f d k h p v w t m

iii) Inorder traversal




i.e. b d f h k m p t v w

Out of which $2^h$ are leaves.

$$\text{Hence internal vertices} = (2^{h+1}-1)-2^h$$
$$= 2^h(2-1)-1$$
$$= 2^h-1$$

**Q.75 Find the maximum of possible height of a binary tree with 13 vertices and draw graph.**

**Ans. :**

We have $n = 13$

The maximum possible height of the binary tree is

$$\frac{n-1}{2} = \frac{13-1}{2} = 6$$

The required graph as shown in Fig. Q.75.1.

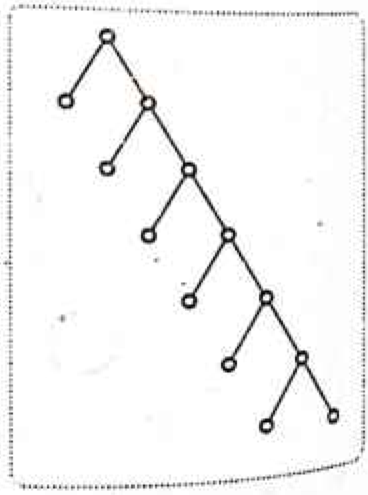


Fig. Q.75.1

**Q.76 Define prefix code and binary search trees.**

[SPPU : Dec.-12, 14, 15, May-08, 15

**Ans. :**

1) A set of sequences is said to be a **prefix code** if no sequences in the set is a prefix of another sequence in the set.

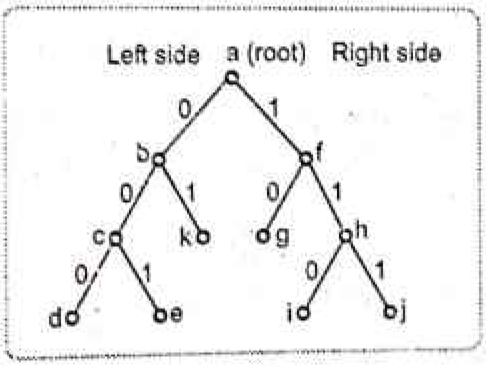
For example the set {000, 001, 01, 10, 11} is a prefix code as no sequence of symbols is present at the beginning of another sequences. All these sequence are distinct.

The set {1, 00, 000, 0001} is not a prefix code because the sequence 00 is the prefix of the sequences 000 and 0001.

A question comes in everyone's mind "How to construct prefix code to a full binary trees?"

We can solve this question by adding some flavours and binary codes to a full binary trees.

For a given full binary tree, we label the two branches incident from each internal node with 0 and 1. For the left branch we assign 0 and for the right branch we assign 1 of every rooted tree or subtree. Consider the following example of full binary tree.




In given figure root a has 2 sons. Left son is ab and right is af, so assign 0 to ab and 1 to af. Now b has two sons. Assign 0 to left son be and 1 to right son bk. Similarly assign 0 or 1 to every edge of a tree.

Now assign to each leaf, a sequence of 0's and 1's which is the sequence of labels of the edges in the path from the root to the leaf.

For example, d is a leaf and the path a to d is a – b – c – d and their respective labels are 0 – 0 – 0 so the prefix code of d is 000.

For leaf e, path is a – b – c – e and labels 0 – 0 – 1

∴ The prefix code of e is 001

Thus the prefix code of above tree is {000, 001, 01, 10, 110, 111}.

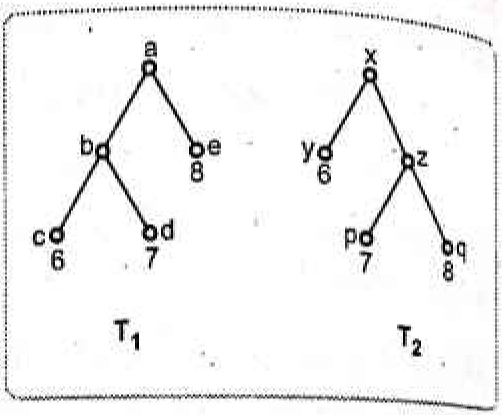
### 2) Optimal Tree

Let T be any full binary tree and $W_1, W_2, W_3, \ldots W_t$ be the weights of the leaves (terminal vertices) then the weight W of the full binary tree is given by

$$W(T) = \sum_{i=0}^{t} W_i l_i$$

Where $l_i = l(i)$ is the length of the path of the leaf i from the root of the tree. The full binary tree is called an optimal tree if its weight is minimum.

For example, suppose 6, 7, 8 are the weights of the leaves in a full binary tree as given below.




In $T_1$, $l(c) = 2$, $l(d) = 2$, $l(e) = 1$

$\therefore$ The weight of T is given by

$$\begin{aligned} W(T_1) &= 6 \times l(c) + 7 \times l(d) + 8 \times l(e) \\ &= 6 \times 2 + 7 \times 2 + 8 \times 1 \\ &= 12 + 14 + 8 = 34 \end{aligned}$$

In $T_2$, $l(y) = 1$, $l(p) = 2$, $l(q) = 2$

$$\begin{aligned} \therefore \quad W(T_2) &= 6 \times 1 + 7 \times 2 + 8 \times 2 \\ &= 6 + 14 + 16 = 36 \end{aligned}$$

Hence $W(T_1) < W(T_2)$

Thus $T_1$ is the optimal tree for the weights 6, 7, 8.

**Q.77 Explain huffman algorithm to find an optimal tree.**

**Ans. :** Let $W_1, W_2, W_3, \ldots W_t$ be the weights of the leaves and it is required to construct an optimal binary tree.

The following steps of an algorithm gives the required optimal binary tree.

**Step 1 :** Arrange the weights in increasing order.

**Step 2 :** Consider two leaves with the minimum weights $W_1$ and $W_2$. Replace these two leaves and their father by a leaf. Assign weight $W_1 + W_2$ to this new leaf.

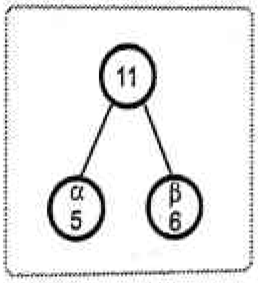
**Step 3 :** Repeat the step 2 for the weights $W_1, W_2, W_3, \ldots W_t$ until no weight remains.

**Step 4 :** The tree obtained in this way is an optimal tree for given weights and stop.

**Q.78 For the following set of weights construct optimal binary prefix code. $\alpha - 5, \beta - 6, \gamma - 6, \delta - 11, \in - 20$** **[SPPU : Dec.-14]**

**Ans. :** Here we use Huffman's coding method.

**Step 1 :** Sort the weights or frequencies of the given letters in increasing order in a queue.




| $\alpha$ | $\beta$ | $\gamma$ | $\delta$ | $\in$ |
|---|---|---|---|---|
| 5 | 6 | 6 | 11 | 20 |
| $\uparrow$ | $\uparrow$ | | | |

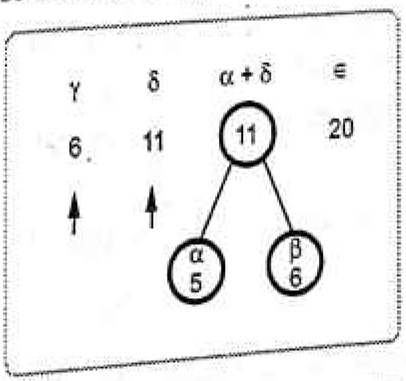
{$\uparrow$ indicates the first 2 smallest weights}

Consider the two symbols with lowest weights say $\alpha, \beta$. The root of the first subtree has a weight

$5 + 6 = 11$.

As $6 < 11$, it can not be placed at the beginning. It has to be placed an appropriate position means after $\delta$. The first subtree is given below

**Step 2 :** Again rewrite sequence of weights in increasing order by replacing and $\beta$ by new subtree weight as

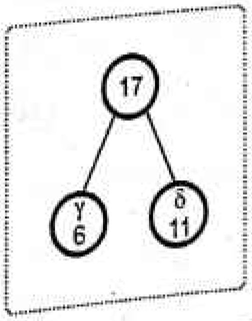



The first two smallest weights are 6 and 11.

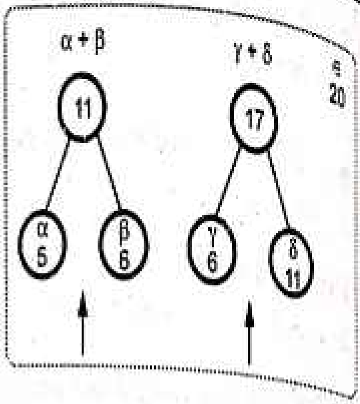
$\therefore$ The root of new subtree has a weight

$$6 + 11 = 17$$

The new subtree is as follows



**Step 3 :** Rewrite sequence of weight in increasing order as

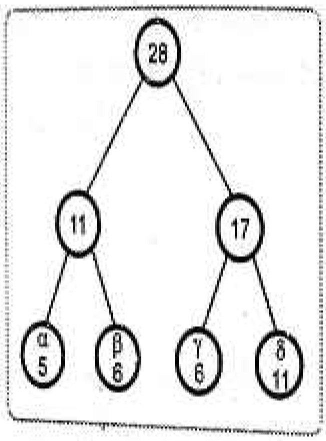



The first two smallest weights are 11 and 17.

∴ The root of new subtree has a weight

$$11 + 17 = 28$$

The new subtree is as follows



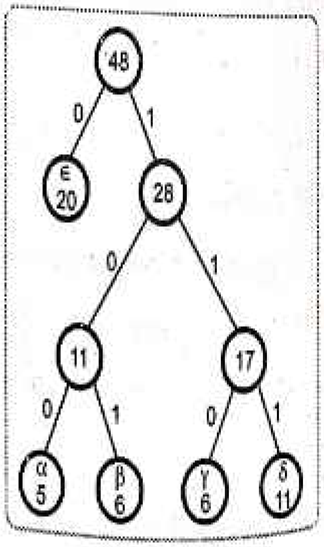

**Step 4 :** Rewrite the sequence of weights in increasing order

$$\begin{array}{cc} \in & \\ 20 & 28 \\ \uparrow & \uparrow \end{array}$$

The root of new subtree is $20 + 28 = 48$

∴ The new subtree is as follows

This is the optimal binary tree.




| Symbols leaves | α | β | γ | δ | ∈ |
|---|---|---|---|---|---|
| **Binary prefix code** | 100 | 101 | 110 | 111 | 0 |

The weight of the optimal tree is

$$W = 20 \times 1 + 5 \times 3 + 6 \times 3 + 6 \times 3 + 11 \times 3$$
$$= 20 + 5 + 18 + 18 + 33 = \mathbf{104}$$

**Note :** All readers are requested to understand Q.78 properly and then see next examples. Hereafter we have given solutions in shortforms.

**Q.79 Suppose data items A, B, C, D, E, F, G occur in the following frequencies.**

| Data Items | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| Weight | 10 | 30 | 5 | 15 | 20 | 15 | 05 |

**Construct a Huffman code for the data. What is the minimum weighted path length.** **☞[SPPU : May-08, Dec.-14]**

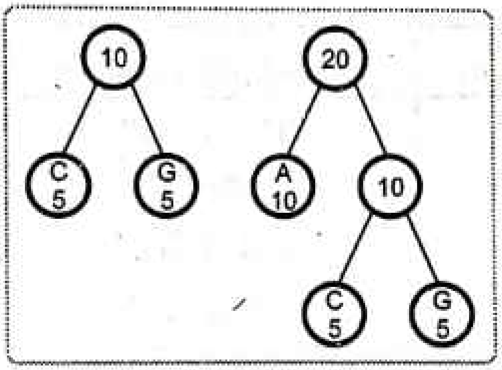
**Ans. :** Consider the following steps

**Step 1 :** Sequence in increasing order is as follows

C D A D F E B

5 5 10 15 15 20 30

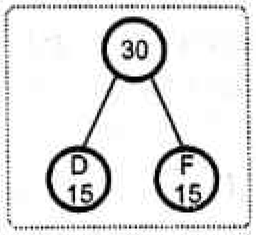
↑ ↑



**Step 2 :** Sequence : D F F

15, 15, 20, 20, 30

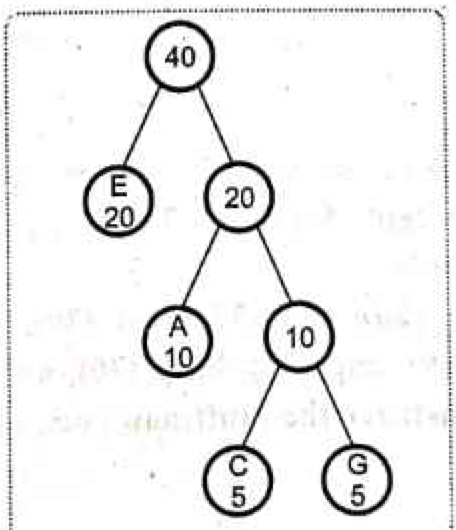
↑ ↑



**Step 3 :** Sequence : E B

20, 20, 30, 30

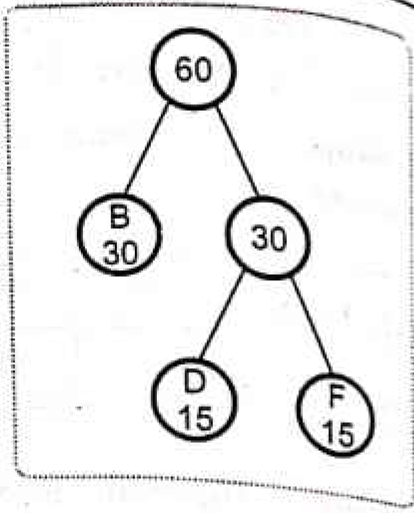
↑ ↑



**Step 4 :** Sequence : B

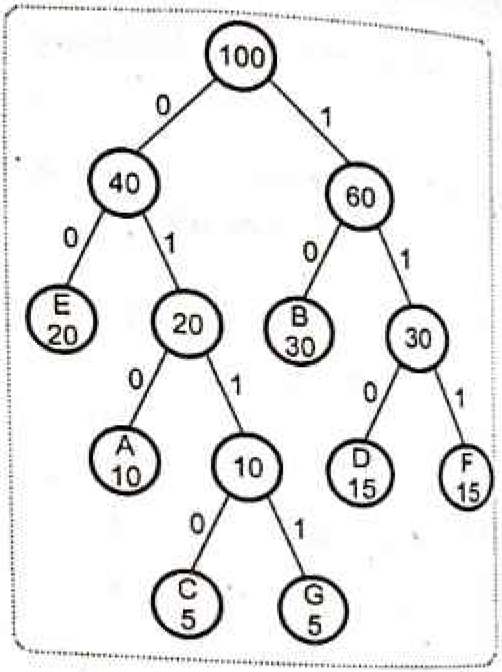
30, 30, 40

↑ ↑




**Step 5 :** Sequence : 40, 60

↑ ↑




| Items | Binary prefix code |
|---|---|
| 5 | 0111 or 0110 |
| 10 | 010 |
| 15 | 110 or 111 |
| 20 | 00 |
| 30 | 11 |

The minimum weight path length for the vertices as follows

$A \to 3$, $B \to 2$, $C \to 4$, $D \to 3$,
$E \to 2$, $F \to 3$, $G \to 4$

∴ The minimum weight of tree is

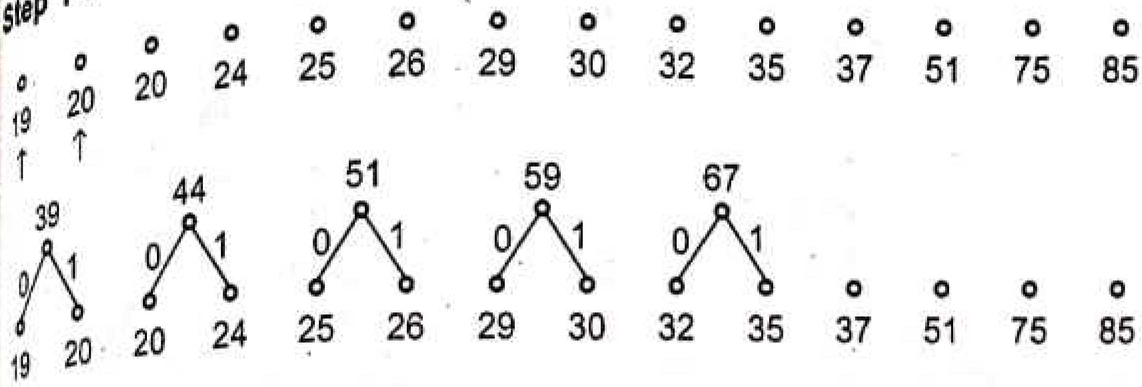
$$W = 10 \times 3 + 30 \times 2 + 5 \times 4 + 15 \times 3$$
$$+ 20 \times 2 + 15 \times 3 + 5 \times 4 = 260$$

**Q.80 A secondary storage media contains information in files with different formats. The frequency of different types of files is as follows.**

**Exe (20), bin (75), bat (20), jpeg (85), dat (51), doc (32), sys (26), c (19), cpp (25), bmp (30), avi (24), prj (29), 1st (35), zip (37). Construct the Huffman code for this.** **[SPPU : May-15, Dec.-15]**

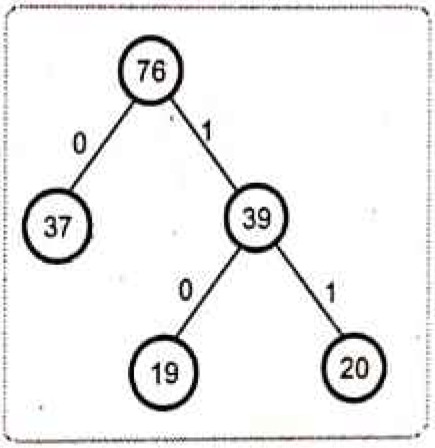
**Ans. :** Consider the following steps

**Step 1 :** Sequence :




**Step 2 :** Sequence :

37, 39, 44, 51, 51, 59, 67, 75, 85

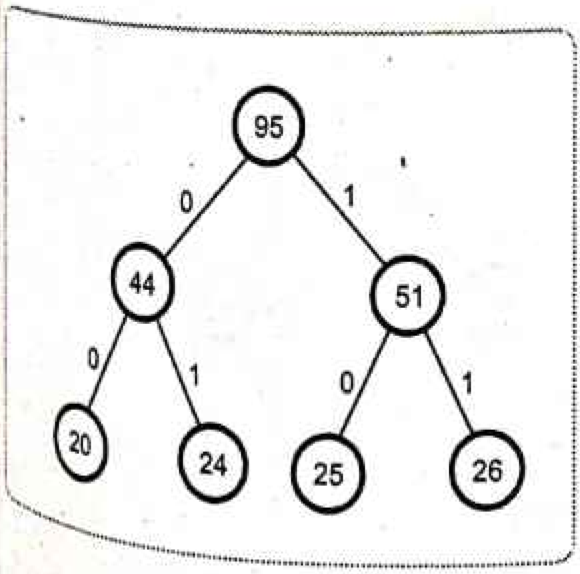


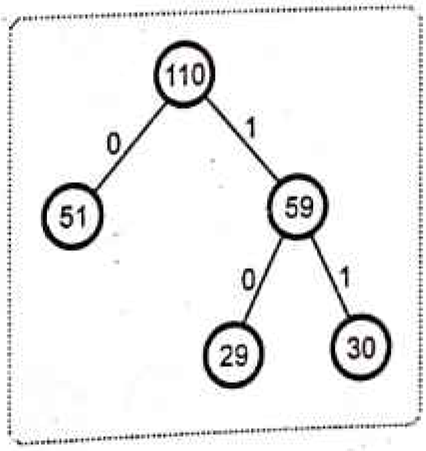

**Step 3 :** Sequence :

44, 51, 51, 59, 67, 75, 76, 85

**Step 4 :** Sequence :

51, 59, 67, 75, 76, 85, 95

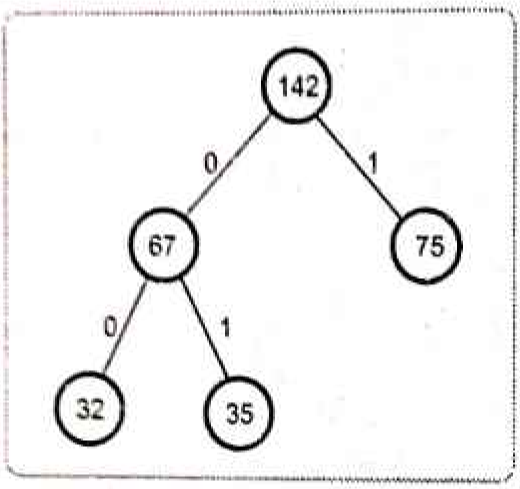


**Step 5 :** Sequence :

67, 75, 76, 85, 95, 100

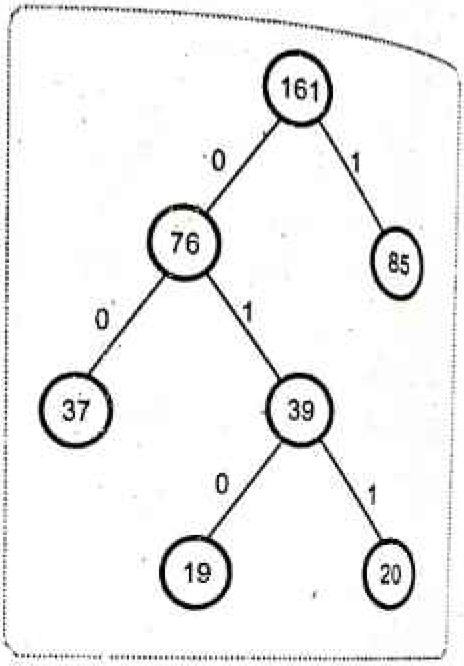
↑ ↑




**Step 6 :** Sequence :

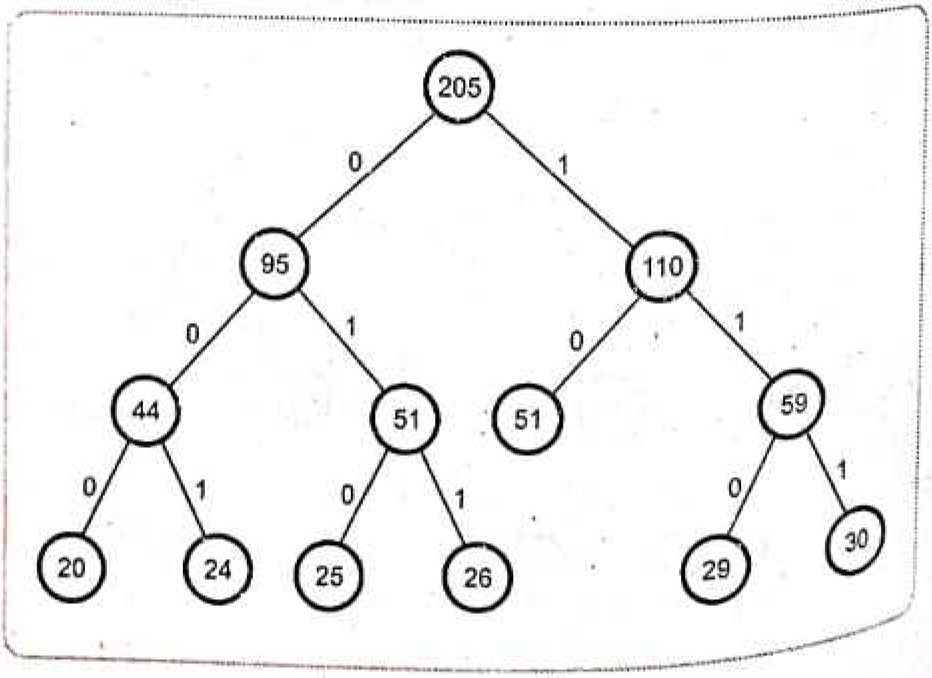
76, 85, 95, 110, 142,

↑ ↑

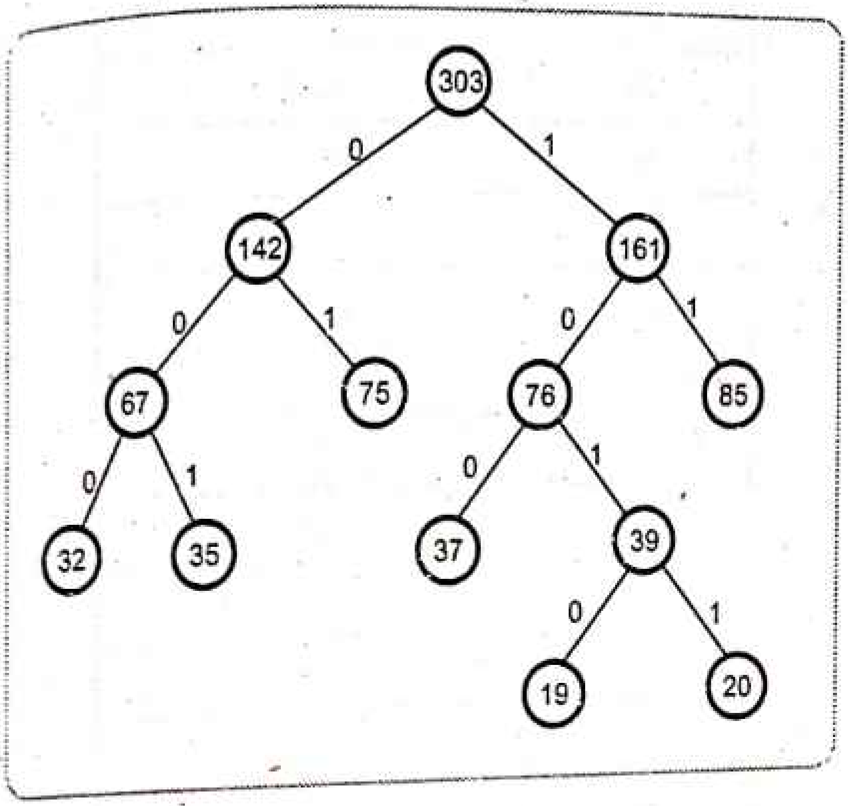



**Step 7 :** Sequence : 95, 110, 142, 161,

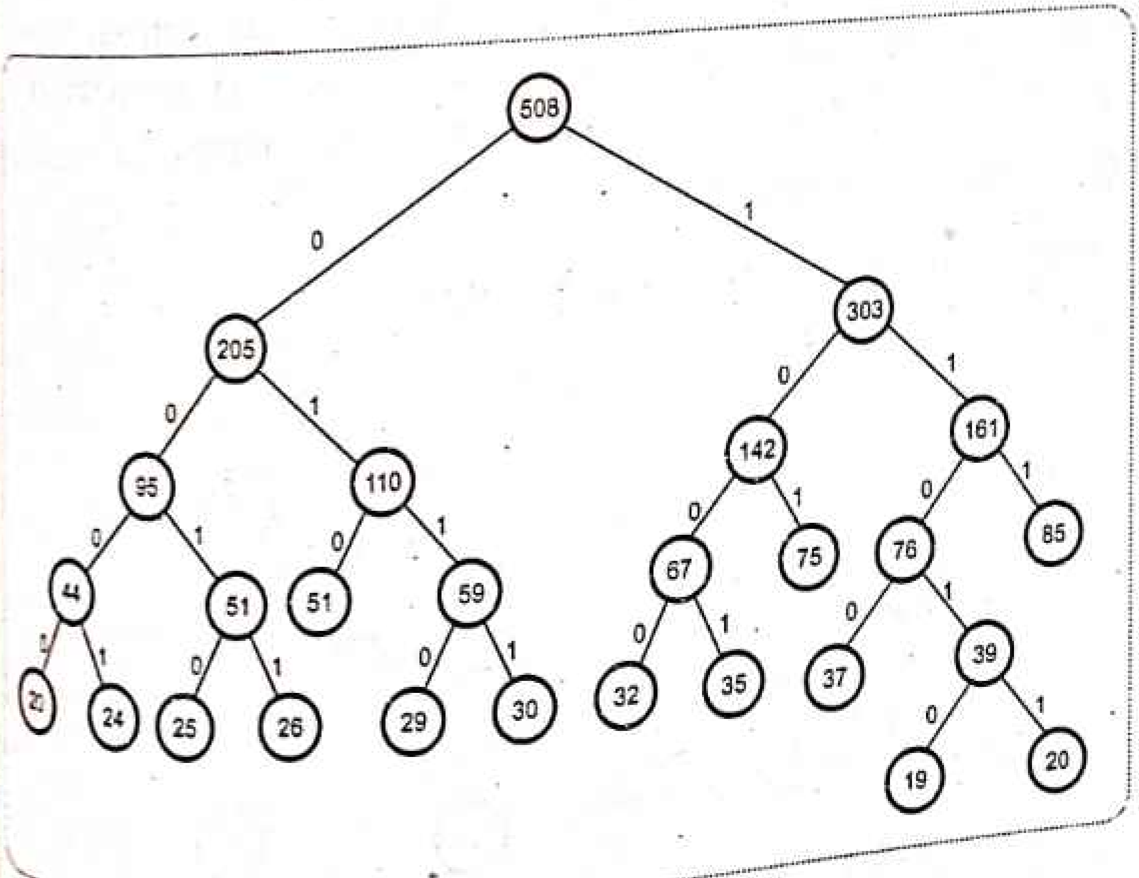
↑ ↑



**Step 8 :** Sequence : 142, 161, 205
$\uparrow \quad \uparrow$




**Step 9 :** Sequence : 205, 303,
$\uparrow \quad \uparrow$

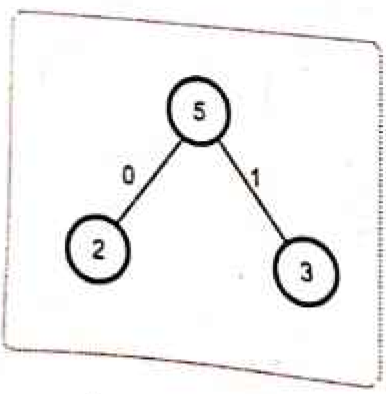


| Numbers | Binary prifix codes |
|---|---|
| 20 | 0000 |
| 24 | 0001 |
| 25 | 0010 |
| 26 | 0011 |
| 51 | 010 |
| 29 | 0110 |
| 30 | 0111 |
| 32 | 1000 |
| 35 | 1001 |
| 75 | 101 |
| 37 | 1100 |
| 19 | 11010 |
| 20 | 11011 |
| 85 | 111 |

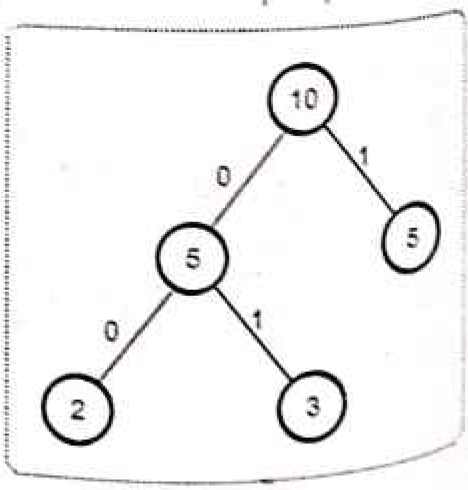
**Q.81** For the following sets of weights, construct an optimal binary prefix code. For each weight in the set, give the corresponding code word i) 2, 3, 5, 7, 9, 13 ii) 8, 9, 10, 11, 13, 15, 22 [SPPU : Dec.-1

**Ans. :** Consider the following steps.

**i) Step 1 :** Sequence :
2, 3, 5, 7, 9, 13
↑ ↑



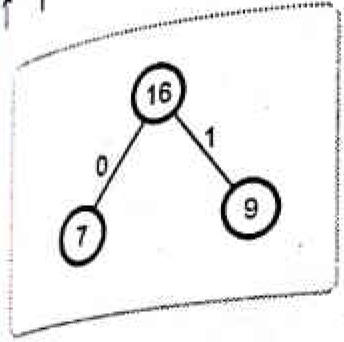

**Step 2 :** Sequence : 5, 5, 7, 9, 13
↑ ↑



**Step 3 :** Sequence :

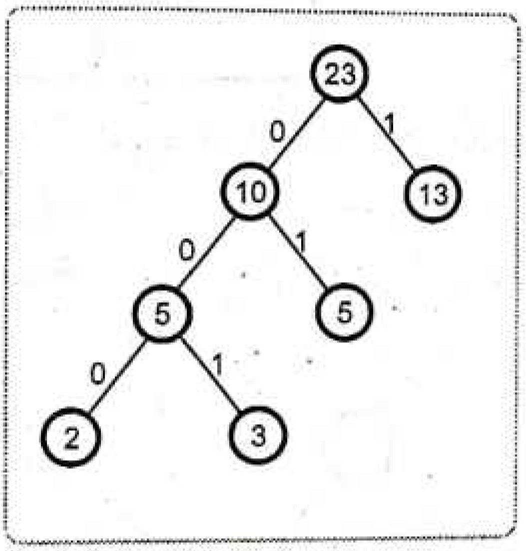
7, 9, 10, 13

↑ ↑



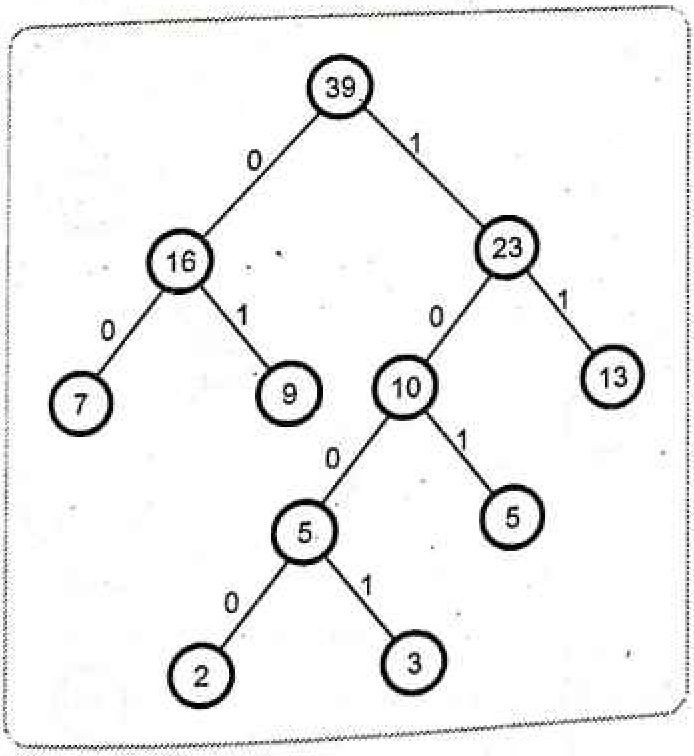

**Step 4 :** Sequence : 10, 13, 16,

↑ ↑




**Step 5 :** Sequence : 16, 23

↑ ↑




| Symbols | Binary prefix code |
|---|---|
| 7 | 00 |
| 9 | 01 |
| 2 | 1000 |

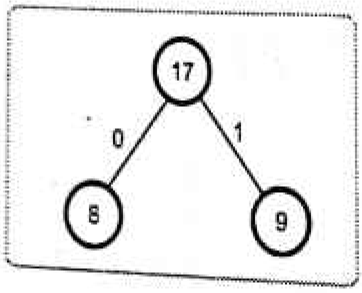
| | |
|---|---|
| 3 | 1001 |
| 5 | 101 |
| 13 | 11 |

ii) Consider the following steps

**Step 1 :** Sequence :

8, 9, 10, 11, 13, 15, 22

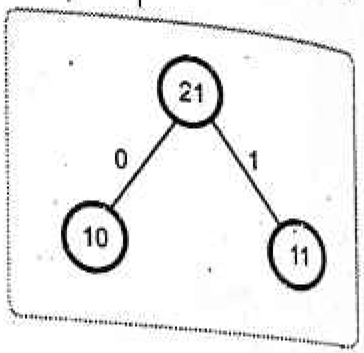
↑ ↑




**Step 2 :**

Sequence : 10, 11, 13, 15, 17, 22

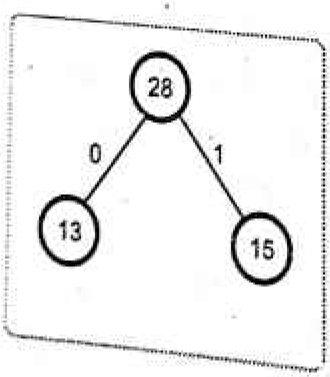
↑ ↑




**Step 3 :** Sequence :

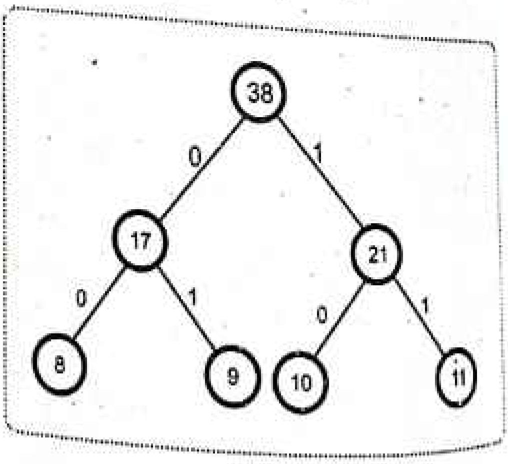
13, 15, 17, 21, 22,

↑ ↑



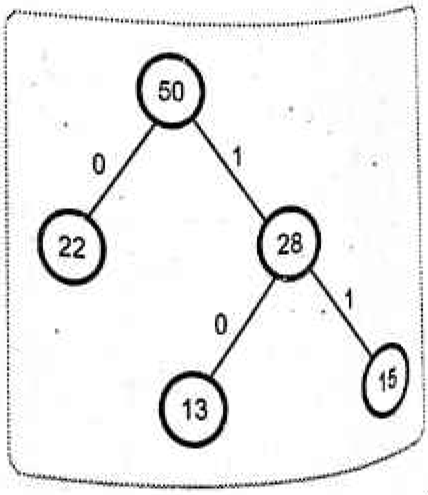

**Step 4 :** Sequence : 17, 21, 22, 28,

↑ ↑

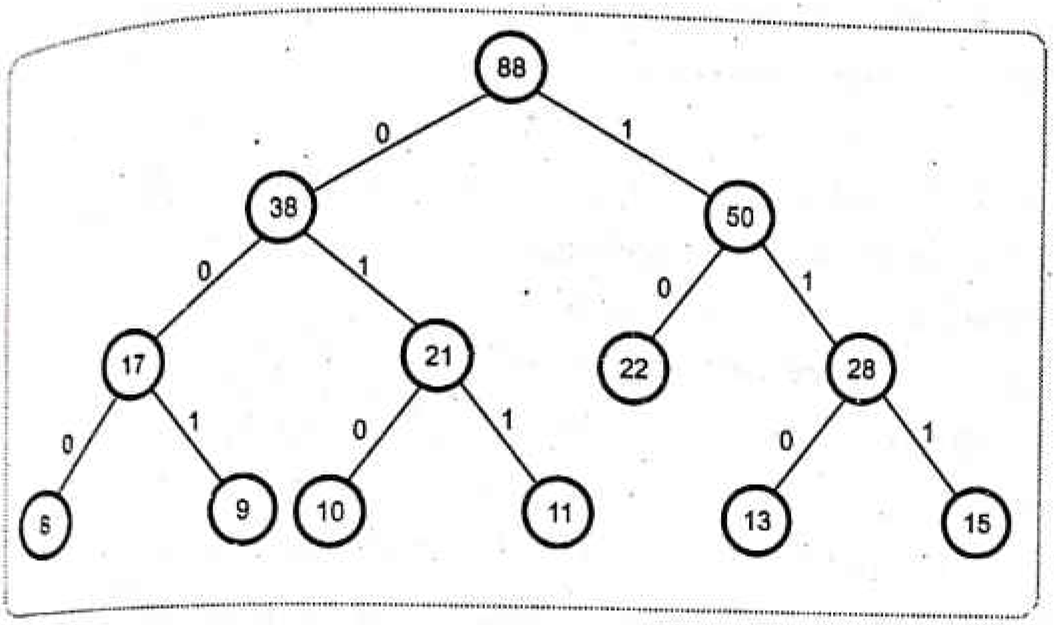



**Step 5 :** Sequence : 22, 28, 38

↑ ↑



**Step 6 :** Sequence : 38, 50
↑ ↑




| Symbols | Binary prefix code |
|---|---|
| 8 | 000 |
| 9 | 001 |
| 10 | 010 |
| 11 | 011 |
| 22 | 10 |
| 13 | 110 |
| 15 | 111 |

## 3.8 : Spanning Tree

**Q.82 Define spanning trees.**

**☞[SPPU : Dec.-11, 13, 14, 15, May-05, 14, 15]**

**Ans. :** A spanning subgraph T of a connected graph G is said to be **spanning tree** of G if T is a tree. In other words, A subgraph T of a graph G is said to be spanning tree if V(T) = V(G).

**Q.83 Explain Prim's algorithm.**

**Ans. :** Let G(V, E) be a connected weighted graph.

To construct minimum spanning tree of G consider the following steps.

**Step 1 :** Select any vertex $v_0$ in graph G.

Set $T = \{v_0, \phi\}$

**Step 2 :** Find edge $e_i = (v_0, v_i)$ in E such that its one end vertex is $v_0 \in T$ and its weight is minimum.

$\therefore$ New set $T = \{\{v_0, v_1\}, \{e_i\}\}$

**Step 3 :** Choose next edge $e_k = (v_k, v_j)$ in such a way that it's one end vertex $v_k \in T$ and other vertex $v_j \notin T$ and weight of $e_k$ is as small as possible. Again join vertex $v_j$ and edge $e_k$ to T.

**Step 4 :** Repeat the step 3 until T contains all the vertices of G. The set T will give the minimum spanning tree of the graph G.

**Q.84 Explain kruskal algorithm.**

**Ans. :** Let G(V, E) be a weighted connect graph.

Consider the following steps.

**Step 1 :** Pick up an edge $e_i$ of G such that its weight $W(e_i)$ is minimum.

(If there are more edges of the minimum weight then select all those edges which do not form a circuit).

**Step 2 :** If edges $e_1, e_2, \ldots, e_n$ have been chosen then pick an edge $e_{n+1}$ such that

i) $e_{n+1} \neq e_i$ for $i = 1, 2, \ldots n$

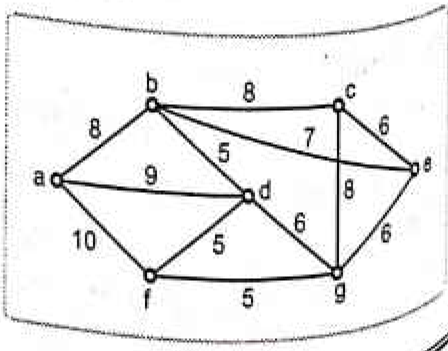
ii) The edges $e_1, e_2, \ldots, e_n, e_{n+1}$ do not form a circuit.

iii) $W(e_{n+1})$ is as small as possible subject to condition (ii).

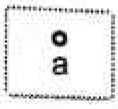
**Step 3 :** Stop when step two cannot be implemented.

**Q.85 Find the minimum spanning tree for the graph given in the following figure using Prim's algorithm.**

**[SPPU : Dec.-14]**

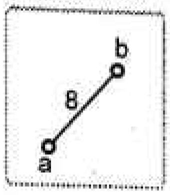


Ans. : Consider the following steps for the construction of the minimum spanning tree.



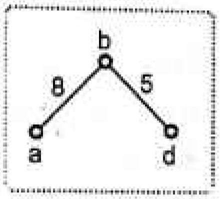

**Step 1 :** Select $a \in G$ as a starting vertex

$\therefore \quad T = \{\{a\}, \phi\}$




**Step 2 :** Vertex a is adjacent to vertices b, d and f. Among these edges minimum weight is $\{a, b\} = 8$

$\therefore \quad T = \{\{a, b\}, \{e_1\}\}$

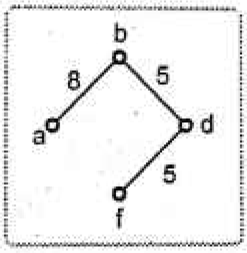



**Step 3 :** Vertex a is adjacent to f, d.

Vertex b is adjacent to c, d, e

The minimum weight is of an edge $\{b, d\} = 5$

$\therefore \quad T = \{\{a, b, d\}, \{e_1, e_2\}\}$



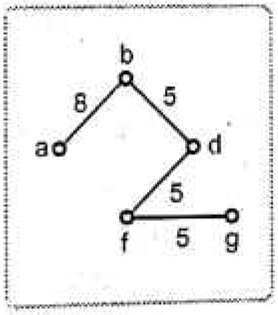

**Step 4 :** Vertex a is adjacent to f and $af = 10$

Vertex b is adjacent to c, e and $bc = 8$, $be = 7$

Vertex d is adjacent to f and g and $df = 5$, $dg = 6$

Among all these weights minimum is $5 = df$

$\therefore \quad T = \{\{a, b, d, f\}\{e_1, e_2, e_3\}\}$




**Step 5 :** Vertex is adjacent to b, d, f but all are in T.

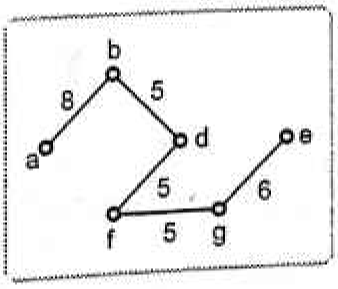
Vertex b is adjacent to c and e

Vertex d is adjacent to g.

Vertex f is adjacent to g

Among all these edges minimum weight is of fg.

$\therefore \quad T = \{\{a, b, d, f, g\}\{e_1, e_2, e_3, e_4\}\}$



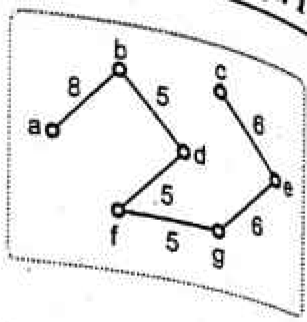

**Step 6 :** Vertex g is adjacent to c and e

Vertex b is adjacent to c and e

Among these edges the minimum weight is $ge = 6$

$\therefore T = \{\{a, b, d, f, g, e\}\{e_1, e_2, e_3, e_4, e_5\}\}$

**Step 7 :** Now only one vertex is remaining.



The vertex C is adjacent to b, e, g.

The minimum weight is ec = 6

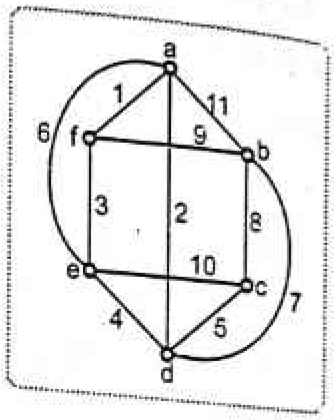
$\therefore \qquad T = \{\{a, b, d, f, g, e, c\}\{e_1, e_2, e_3, e_4, e_5, e_6\}\}$

The graph obtained is the minimum spanning tree of weight

$$= 8 + 5 + 5 + 5 + 6 + 6 = 35$$

**Q.86 Determine minimum spanning tree for the given graph by Prism's algorithm.**

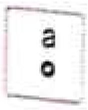
**☞[SPPU : May-14]**



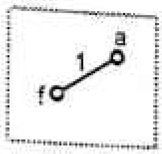
**Ans. :** Consider the following steps for the construction of the minimum spanning tree.

**Step 1 :** Starting with vertex a

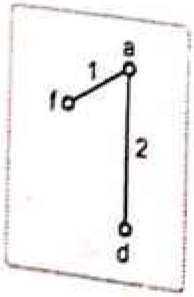
$T = \{\{a\}, \phi\}$



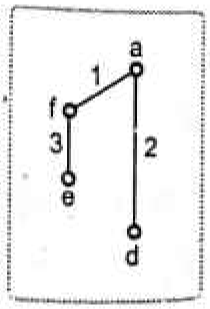
**Step 2 :** $T = \{\{a, f\}, e_1\}$



**Step 3 :** $T = \{\{a, f, d\}, \{e_1, e_2\}\}$

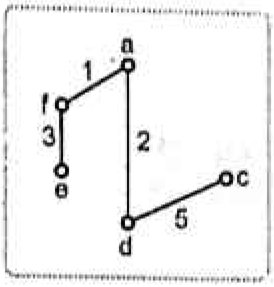


**Step 4 :** $T = \{\{a, f, d, e\}, \{e_1, e_2, e_3\}\}$



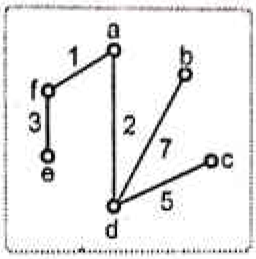
**Step 5 :**

$T = \{\{a, f, d, e, c\}, \{e_1, e_2, e_3, e_4\}\}$




**Step 6 :**

$T = \{\{a, f, d, e, c, b\}, \{e_1, e_2, e_3, e_4, e_5\}\}$

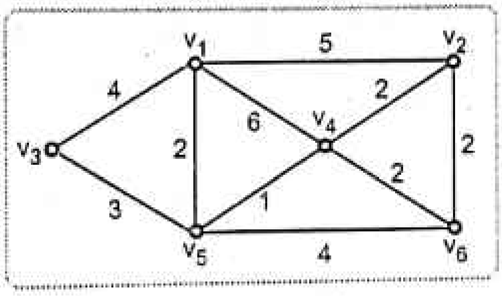



The graph obtained is the minimum spanning tree of weight

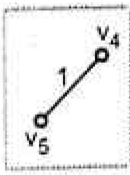
$$= 1 + 2 + 1 + 5 + 7 = 18$$

**Q.87 Find the minimum spanning tree for graph given below by Kruskal's algorithm.**

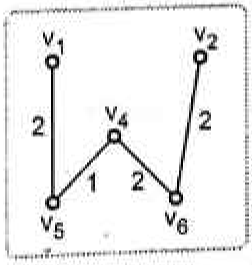
**☞[SPPU : Dec.-13]**



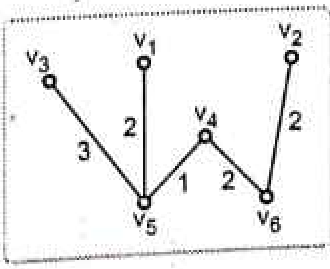

**Ans. :** Consider the following steps for the construction of the minimum spanning tree.




**Step 1 :** The minimum weight in given graph is 1, so select an edge $\{v_4, v_5\}$




**Step 2 :** The minimum weight is 2 in remaining graph. There are four edges of weight 2. These edges form a circuit so select edges which do not form a circuit with selected edge. We select three edges $\{v_5, v_1\}, \{v_4, v_6\}, \{v_6, v_2\}$

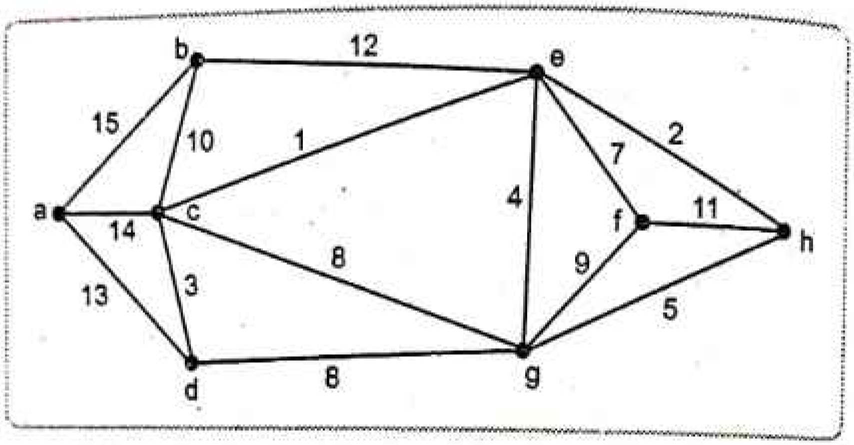



**Step 3 :** The minimum weight is 3 in the remaining graph so select $\{v_5, v_3\}$

The graph obtained is the minimum spanning tree of weight

$$= 3 + 2 + 1 + 2 + 2 = 10$$

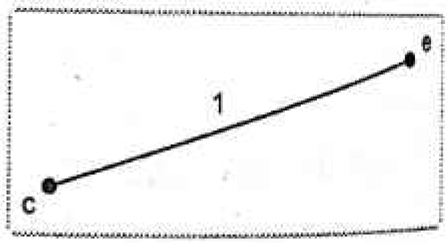
**Q.88 Obtain the minimum spanning tree for the following graph. Obtain the total cost of minimum spanning tree.**



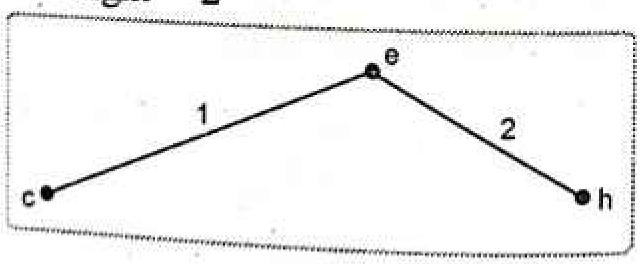

☞[SPPU : May-15, Dec.-15

**Ans. :** Using Kruskal algorithm, the minimum spanning tree is obtained as follows :

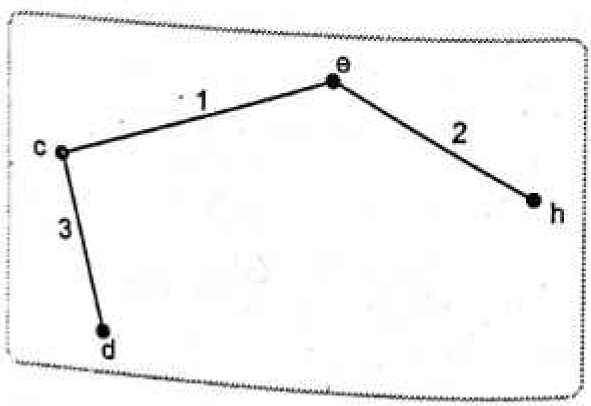
**Step 1 :** Minimum weight in a given graph is 1 associated with edge {a, e}



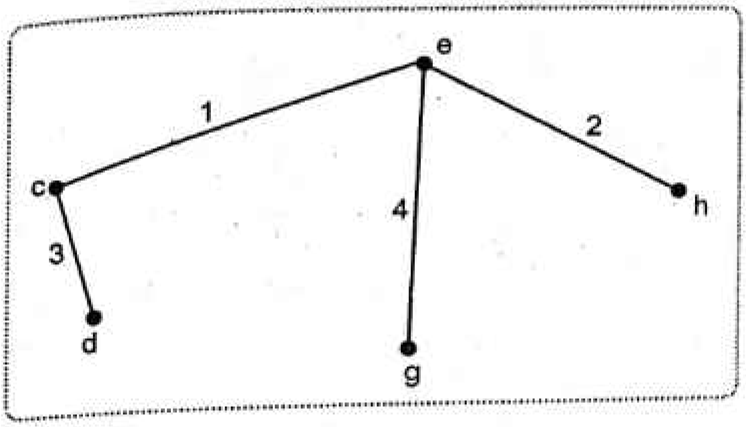

**Step 2 :** Minimum weight = 2



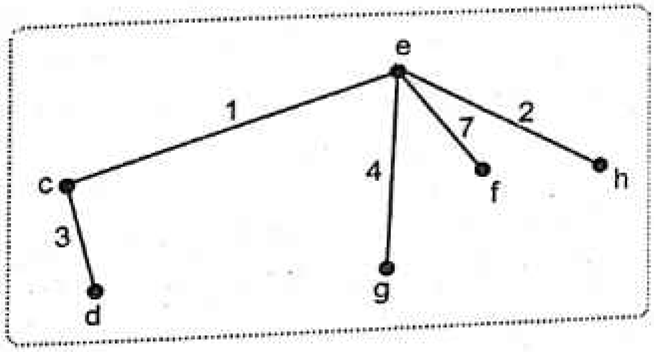

**Step 3 :** Minimum weight = 3



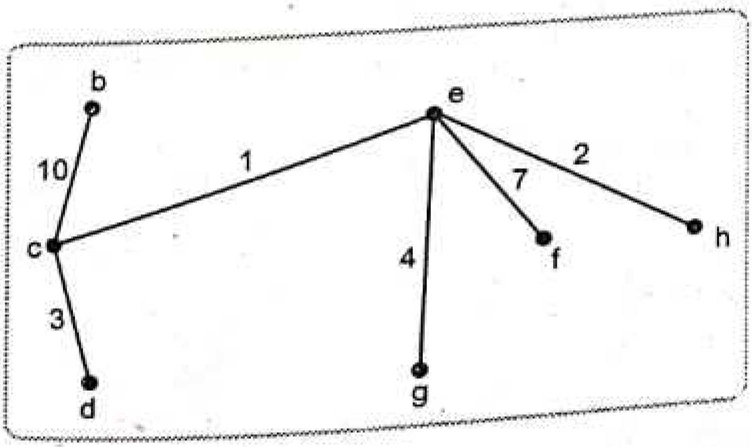
**Step 4 :** Minimum weight = 4



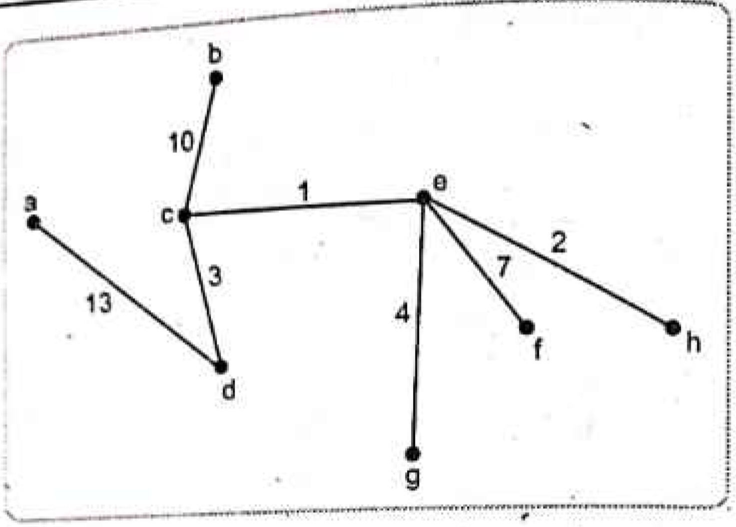

**Step 5 :** Minimum weight = 7




**Step 6 :** Minimum weight = 10




**Step 7 :** Minimum weight = 13




The obtained graph is the minimum spanning tree of the given graph. Its total cost is 40.

## 3.9 : Fundamental Cutset and Circuits

**Q.89 Explain fundamental circuits and cutsets with examples.**

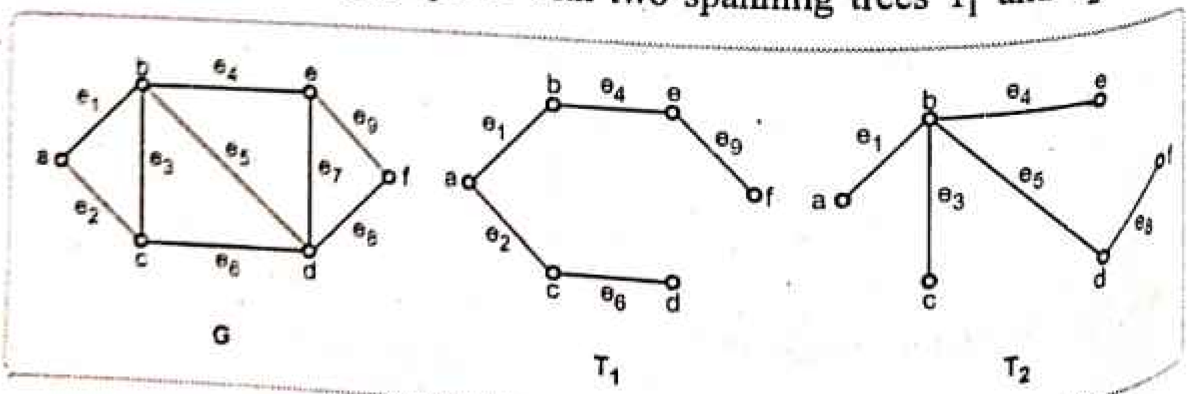
**[SPPU : Dec.-07, 12, 14, 15, May-07, 15]**

**Ans. :** Let G be a connected graph and T be a spanning tree of G. An edge of a tree is called a **branch.** An edge of G which is not in T is called chord of T. T + e contains a unique circuit called fundamental circuit of G with respect to T.

A fundamental circuit of a connected graph G is always with respect to a spanning tree of G.

Therefore the different spanning trees will have different fundamental circuits.

Consider the following graph G with two spanning trees $T_1$ and $T_2$.




In graph G, vertex set V(G) = {a, b, c, d, e, f}

Edge set E(G) = $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$

### A) For Spanning Tree $T_1$ :

Branches of $T_1$ are $e_1, e_2, e_4, e_6, e_9$

Chords of $T_1$ are $e_3, e_5, e_4, e_8$

Consider the following table of chords and corresponding fundamental circuits.

| Chords | Corresponding Fundamental Circuits |
|---|---|
| $e_3$ | $\{e_1, e_2, e_3\}$ |
| $e_5$ | $\{e_1, e_2, e_6, e_5\}$ |
| $e_7$ | $\{e_1, e_2, e_4, e_6, e_7\}$ |
| $e_8$ | $\{e_1, e_2, e_4, e_6, e_9, e_8\}$ |

### B) For Spanning Tree $T_2$

Branches of $T_2$ are $e_1, e_3, e_5, e_4, e_8$

Chords of $T_2$ are $e_2, e_4, e_6, e_7, e_9$

Consider the following table of chords and corresponding fundamental circuits.

| Chords | Corresponding Fundamental Circuits |
|---|---|
| $e_2$ | $\{e_1, e_3, e_2\}$ |
| $e_6$ | $\{e_3, e_5, e_6\}$ |
| $e_7$ | $\{e_4, e_5, e_7\}$ |
| $e_9$ | $\{e_4, e_5, e_8, e_9\}$ |

### Fundamental Cutsets

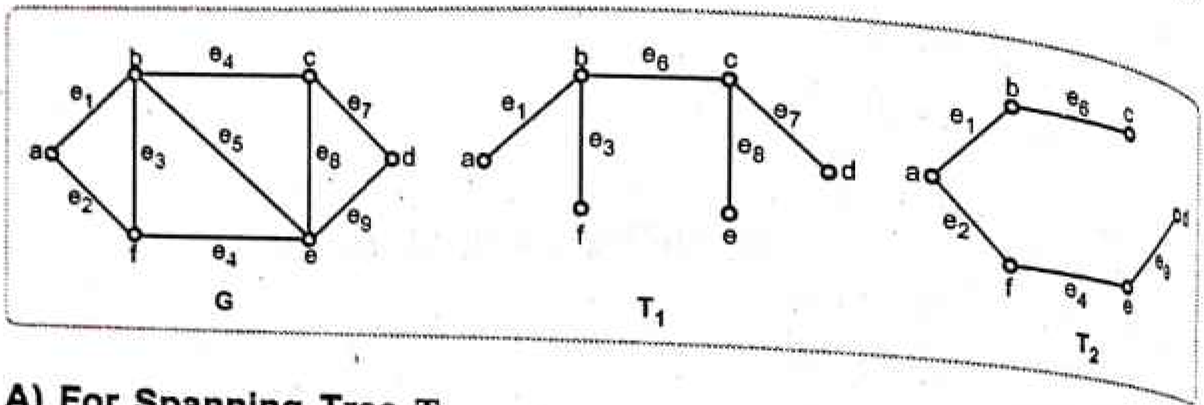
Let T be k spanning tree of a connected graph G. Since every edge e of a tree is an isthmus or bridge, T-e splits into two components say $T_1$ and $T_2$. But

$$V(T) = V(T_1) \cup V(T_2) = V(G)$$

The set E of edges of G which join a vertex in $V(T_1)$ to a vertex in $V(T_2)$ is a cutset of G.

A cutset of G obtained in this manner is called a fundamental cutset of G with respect to T. To each edge of edge there is a fundamental cutset and every fundamental cutset is obtained in this way. Thus the number of fundamental cutsets of G w.r.t. T is the number of branches of T.

**Theorem :** Let G be a connected graph with n vertices then its spanning tree has $n-1$ edges and there are $n-1$ fundamental cutsets only.
e.g. Consider the following and its spanning tree.



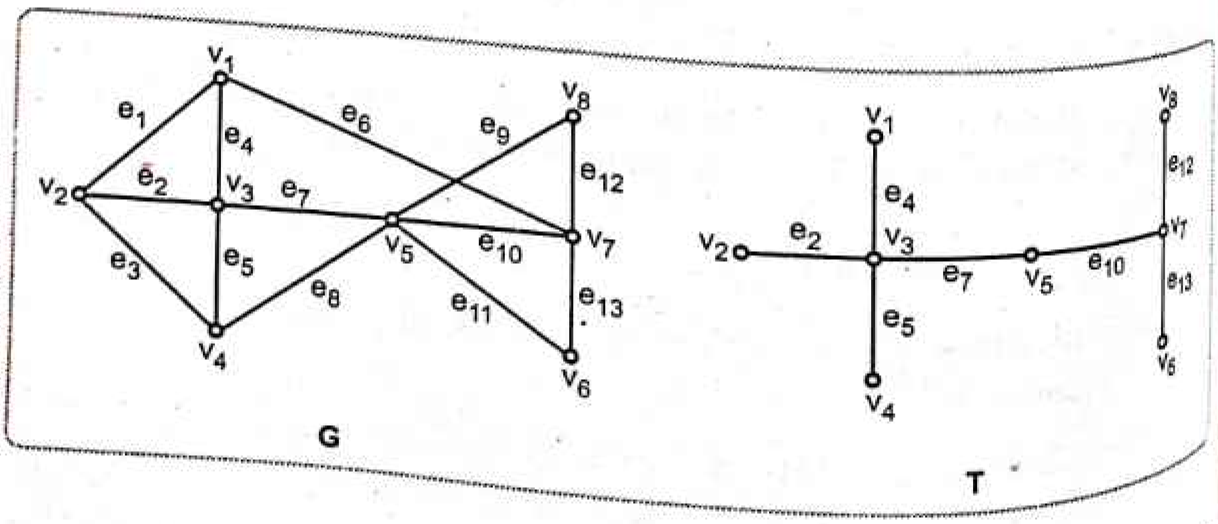

### A) For Spanning Tree $T_1$

Branches of $T_1$ are $e_1$, $e_3$, $e_6$, $e_7$, $e_8$

Consider the following table

| Branches $T_1$ | Corresponding Fundamental Cutset |
|---|---|
| $e_1$ | $\{e_1, e_2\}$ |
| $e_3$ | $\{e_3, e_2, e_4\}$ |
| $e_6$ | $\{e_6, e_5, e_4\}$ |
| $e_7$ | $\{e_7, e_9\}$ |
| $e_8$ | $\{e_8, e_4, e_9, e_5\}$ |

**Q.90 Find the fundamental system of cutset for the graph G shown below w.r.t. the spanning tree T.** **[SPPU : May-15, Dec.-12, 15]**

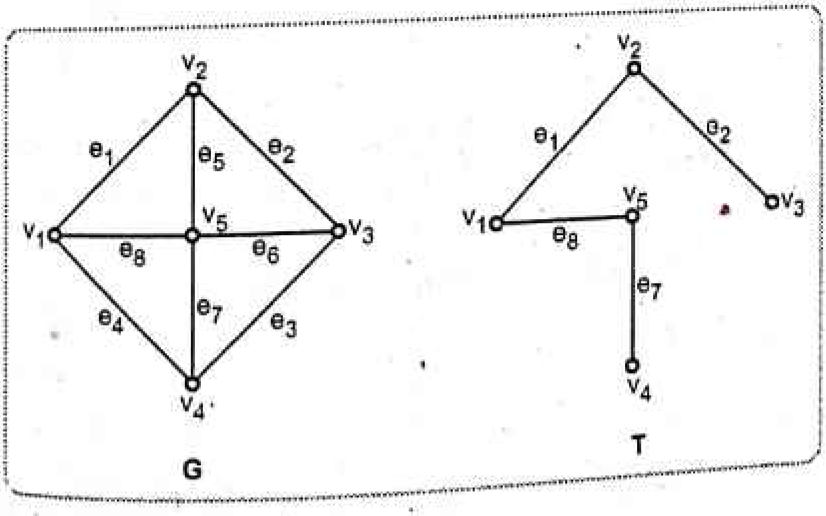


**Ans. :** The spanning tree T has 7 branches $\{e_2, e_4, e_5, e_7, e_{10}, e_{12}, e_{13}\}$. Therefore are seven fundamental cutsets of G w.r.t. T which are given below :

| Branch | Fundamental Cutset |
|---|---|
| $e_2$ | $\{e_2, e_1, e_3\}$ |
| $e_4$ | $\{e_4, e_1, e_6\}$ |
| $e_5$ | $\{e_5, e_3, e_8\}$ |
| $e_7$ | $\{e_7, e_6, e_8\}$ |
| $e_{10}$ | $\{e_{10}, e_6, e_9, e_{11}\}$ |
| $e_{12}$ | $\{e_{12}, e_9\}$ |
| $e_{13}$ | $\{e_{13}, e_{11}\}$ |

**Q.91 Find the fundamental cutsets and fundamental circuits of the following graph w.r.t. given spanning tree.**

**[SPPU : May-07, Dec.-07, 14]**




**Ans. :** Here the spanning tree has 4 branches $e_1, e_2, e_7, e_8$. Therefore there are 4 fundamental cutsets corresponding to each branch of T which are given below.

| Branch | Corresponding Fundamental Cutset |
|---|---|
| $e_1$ | $\{e_1, e_5, e_6, e_3\}$ |
| $e_2$ | $\{e_2, e_6, e_3\}$ |
| $e_7$ | $\{e_7, e_3, e_4\}$ |
| $e_8$ | $\{e_8, e_4, e_5, e_6, e_3\}$ |

The chords of T are $e_3$, $e_4$, $e_5$, $e_6$

Therefore there are 4 fundamental circuits corresponding to each chord of T which are given below :

| Chord | Corresponding Fundamental Cutset |
|---|---|
| $e_3$ | $\{e_1, e_2, e_7, e_8, e_3\}$ |
| $e_4$ | $\{e_4, e_7, e_8\}$ |
| $e_5$ | $\{e_5, e_1, e_8\}$ |
| $e_6$ | $\{e_6, e_1, e_2, e_8\}$. |

## 3.10 : Max Flow-Min Cut Theorem

**Q.92 Explain labelling procedure for finding maximum flow in the Network.**

**Ans. :**

1) The source a is labelled $(-, \infty)$. It means that (out from nowhere) the source can supply can infinite amount of material to the other vertices.

2) A vertex b that is adjacent from a is labelled $(a^+, \Delta b)$, where $\Delta b$ is equal to $w(a, b) - \phi(a, b)$,

if $w(a, b) > \phi(a, b)$ ;

i.e. $$\Delta b = w(a, b) - \phi(a, b) \qquad [\text{if } w(a, b) > \phi(a, b)]$$

The vertex is not labelled if $w(a, b) = \phi(a, b)$

3) Scan and label all the remaining vertices adjacent to a. Also scan and label all the vertices adjacent to labelled vertices.

Suppose vertex q is adjacent to labelled vertex b, then q is labelled as $(b^+, \Delta q)$

where $\Delta q = \min\{\Delta b, [w(b, q) - \phi(b, q)]\}$

if $w(b, q) > \phi(b, q)$

Vertex q is not labelled if $w(b, q) = \phi(b, q)$

We can also label vertex q as $(\bar{b}, \Delta q)$ where

$$\Delta q = \min[\Delta b, \phi(q, b)] \quad \text{if} \quad \phi(q, b) > 0$$

4) Repeat step 3 till we reach to sink z.

5) If we repeat this labelling procedure, two cases shall arise while labelling the sink z.

**Case I :**

i) Sink z is labelled, say with a label $(y^+, \Delta z)$ [z is never labelled $(y^-, \Delta z)$]

ii) Vertex y can be labeled as $(q^+, \Delta y)$ or $(q^-, \Delta y)$ for some adjacent vertex q.

iii) If y is labelled $(q^+, \Delta y)$ ($q^+$ means increase in flow) then we increase the flow in edge (q, y) to $\phi(q, y)$ to $\Delta z$. Similarly for the label $(q^-, \Delta y)$ we decrease the flow in edge (q, y) to $\phi(q, y) - \Delta y$.

iv) This process is continued back to source a till the value of flow is increased by amount $\Delta z$.

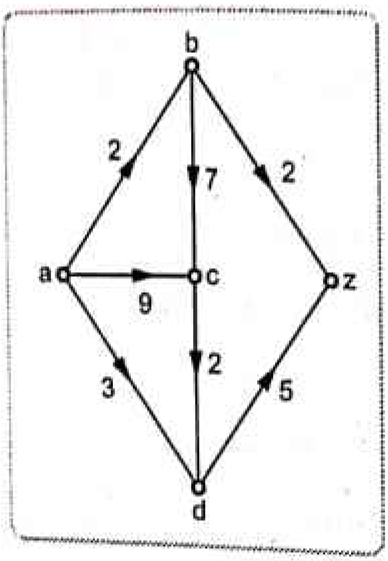
v) Again start the labelling procedure to further increase value of flow in the network.

**Case II :**

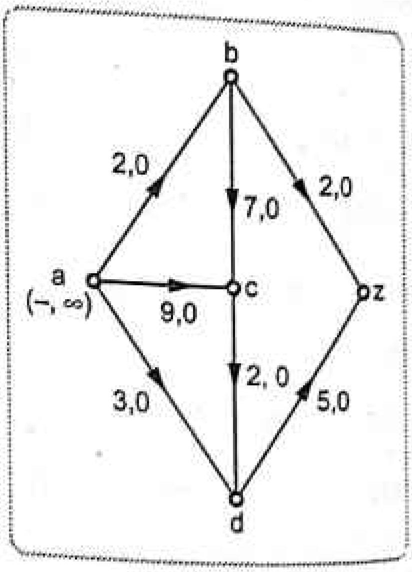
i) If the sink z is not labelled, then denote all labelled vertices as P and all unlabelled vertices as $\bar{P}$.

ii) The fact that sink z is not labeled means flow each edge directed from vertices of P to vertices of $\bar{P}$. is equal to capacity of cut $(P, \bar{P})$ is thus maximum flow.

**Q.93 Determine the maximal flow in the following transport network.** ☞[SPPU : Dec.-12, May-14]




**Ans. : Step 1 :** Assign the flow zero to each edge and the label $(-, \infty)$ to the source a.




**Step 2 :** The vertices b, c, d are adjacent to the source a.

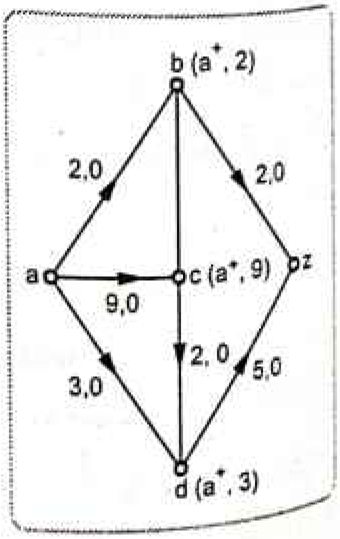
Therefore, we label the vertices b, c, d.

For the vertex b

$$w(a, b) - \phi(a, b) = 2$$

i.e. $\Delta b = 2$

Similarly $\Delta c = 9$ and $\Delta d = 3$




Now sink z is adjacent to both vertices b and d, so we arbitrarily choose vertex d and label.

sin z as $(d^+, 5)$ as

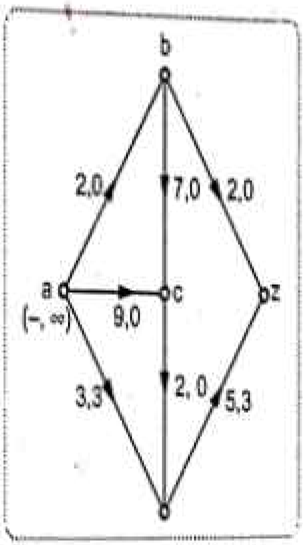
$$w(d, z) - \phi(d, z) = 5 - 0 = 5$$

and $\Delta z =$

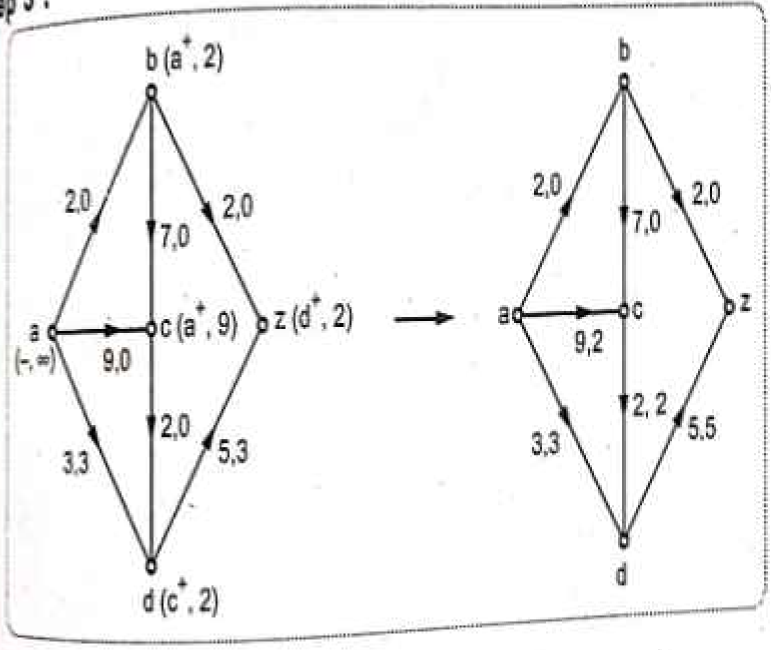
$$\min\{\Delta d, \{w(d, z) - \phi(d, z)\}$$

$$= \min\{3, 5\} = 3$$

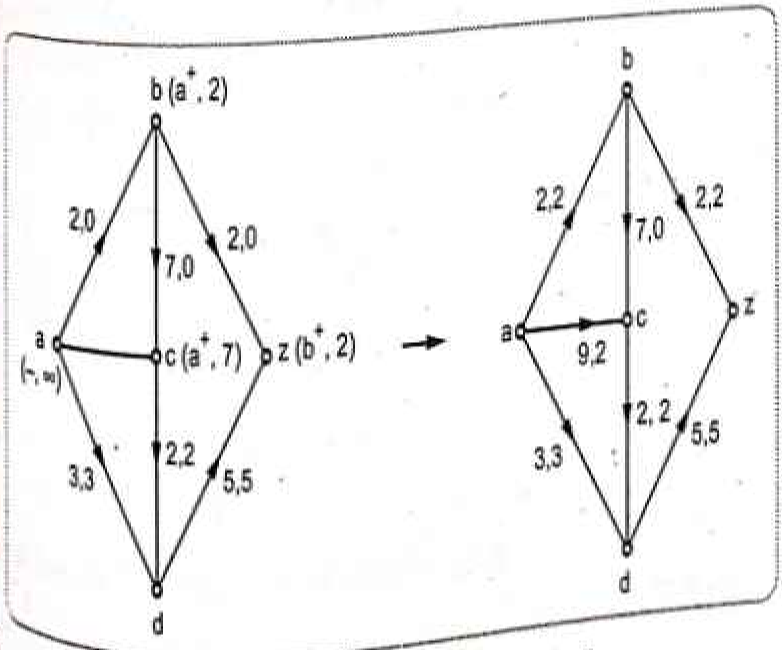
According to label of sink, we now adjust flow of edge (d, z) and edge (a, d)




**Step 3 :**



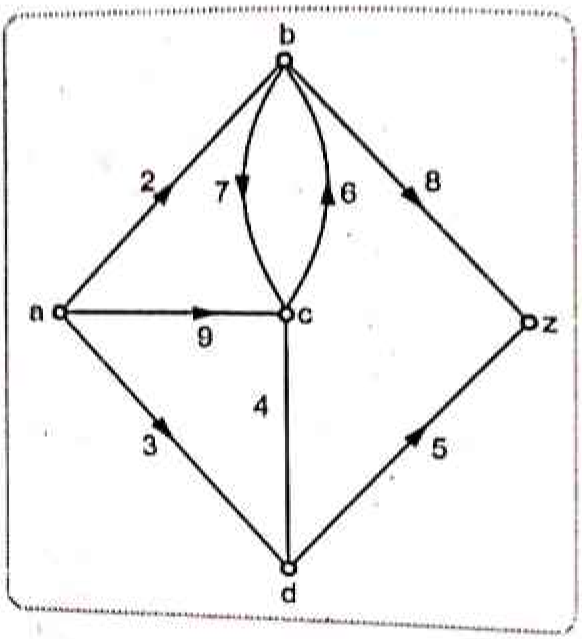

**Step 4 :**



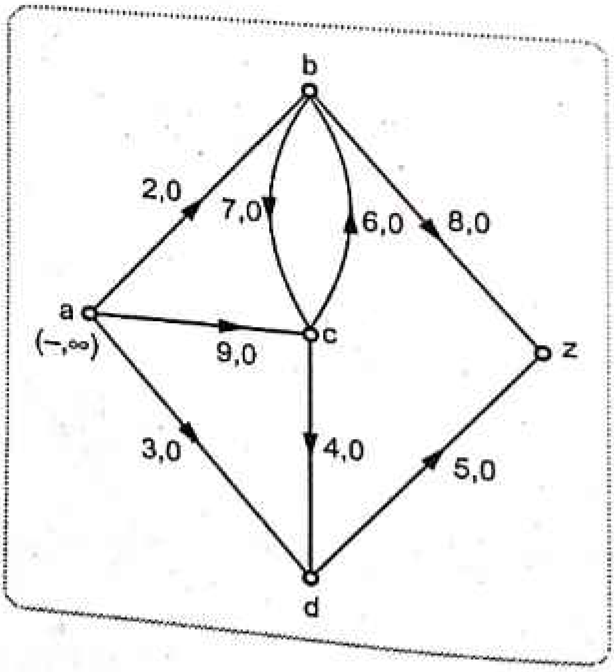

From the step 4, it is clear that the maximum flow is 7.

**Q.94 Determine the maximal flow in the flowing transport network.**

[SPPU : Dec.-13]




**Ans. : Step 1 :** Assign the flow zero to each edge and the label (–, ∞) to the source a.



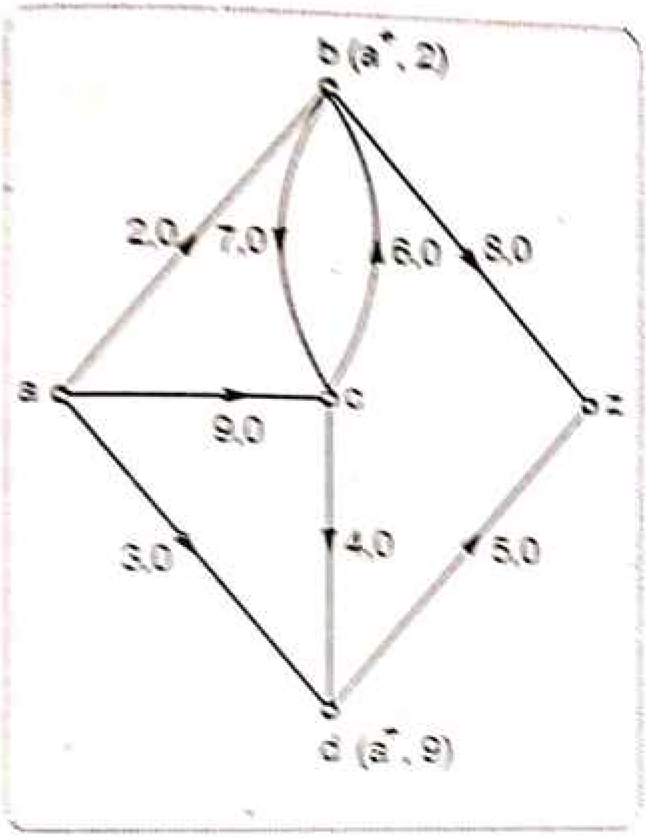

**Step 2 :** The vertices b, c, d are adjacent to the source a. Therefore, we label the vertices b, c, d for the vertex b.

$w(a, b) - \phi(a, b) = 2$

$\Delta b = 2,$

i.e.

Similarly $\Delta c = 9$, $\Delta d = 3$



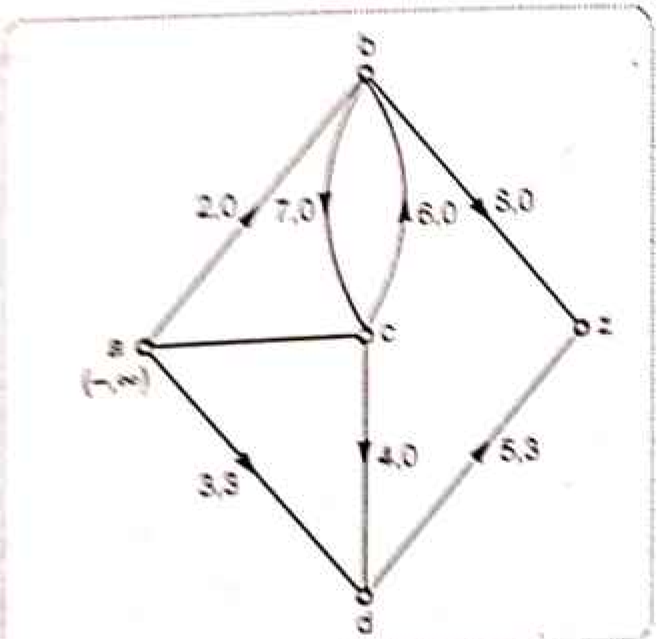

Now sink z is adjacent to both vertices b and d, so we can choose any vertex b or d for labelling of sink z. Let us choose the vertex d. The label of sink z is $(d^-, 5)$ as $w(d, z) - \phi(d, z) = 5 - 0 = 5$

and

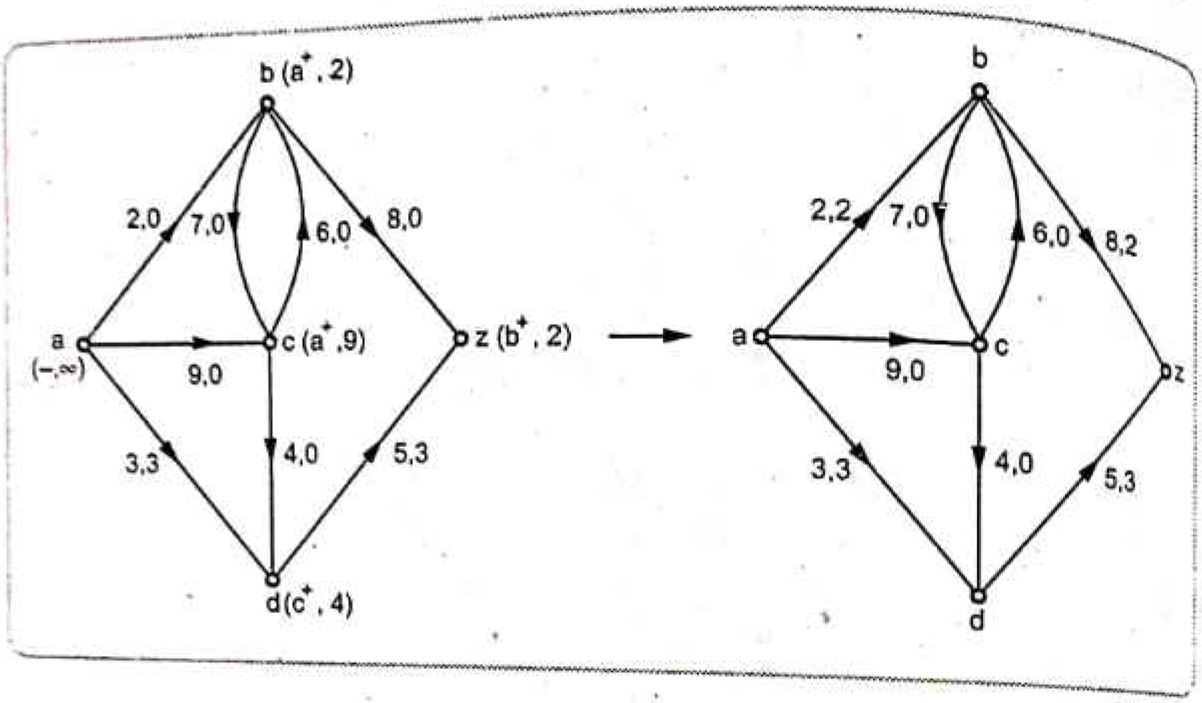
$\Delta z = \min\{\Delta d, w(d, z) - \phi(d, z)\}$

$= \min\{3, 5\} = 3$

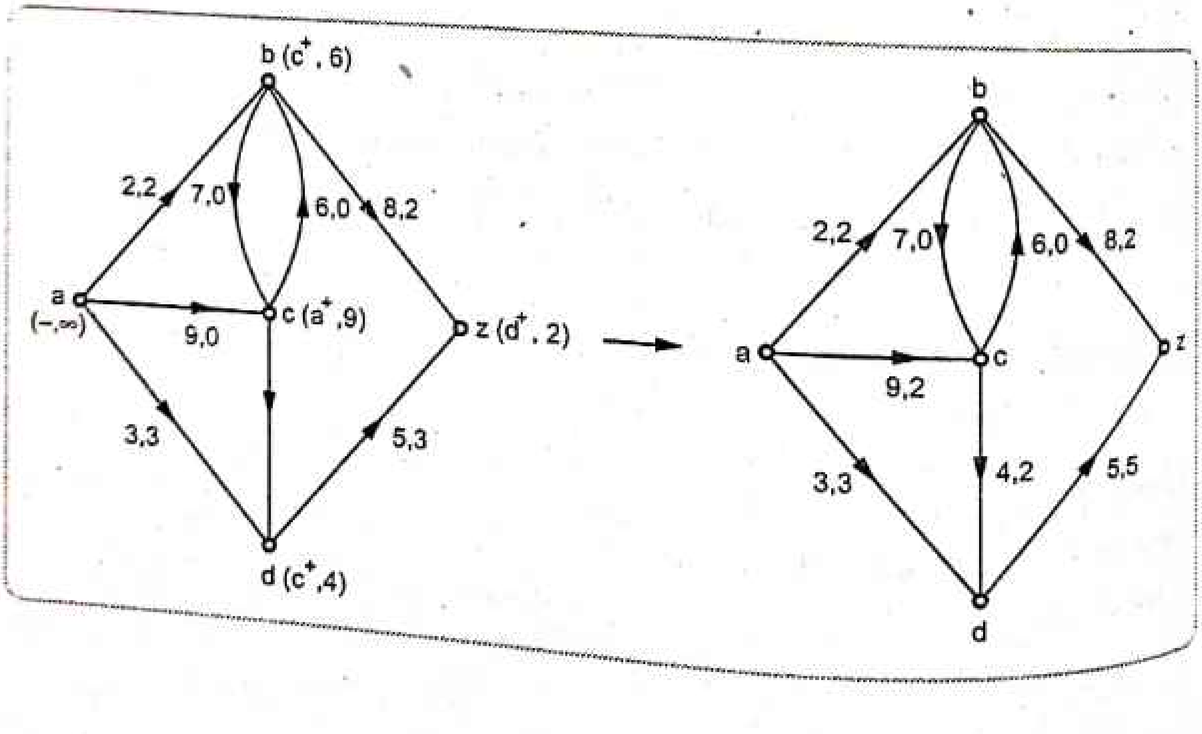
According to label of sink, we now adjust flow of edge (d, z) and edge (a, d)



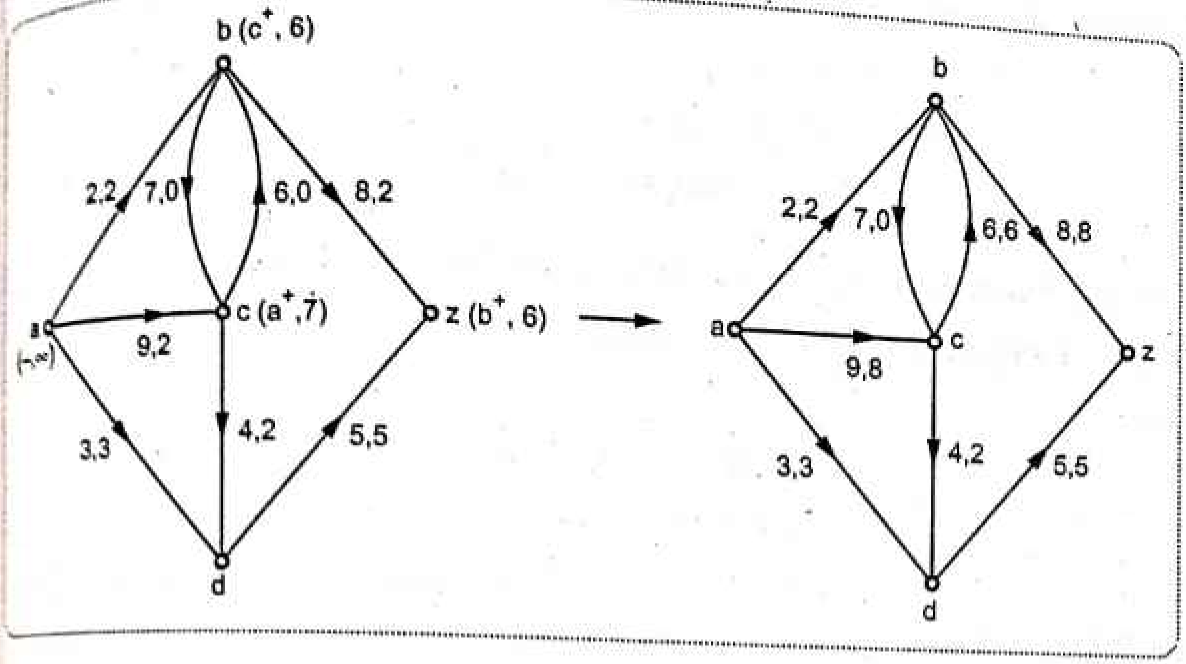
Step 3 :


b (a⁺, 2)
2,0
7,0
6,0
8,0
a
(–,∞)
9,0
c (a⁺,9)
z (b⁺, 2)
3,3
4,0
5,3
d(c⁺, 4)
b
2,2
7,0
6,0
8,2
a
9,0
c
z
3,3
4,0
5,3
d

Step 4 :


b (c⁺, 6)
2,2
7,0
6,0
8,2
a
(–,∞)
9,0
c (a⁺,9)
z (d⁺, 2)
3,3
5,3
d (c⁺,4)
b
2,2
7,0
6,0
8,2
a
9,2
c
z
3,3
4,2
5,5
d

Step 5 :




**Q.95 Use labelling procedure to find a maximum flow in the transport network shown in the following Fig. Q.95.1. Determine the corresponding minimum cut.**

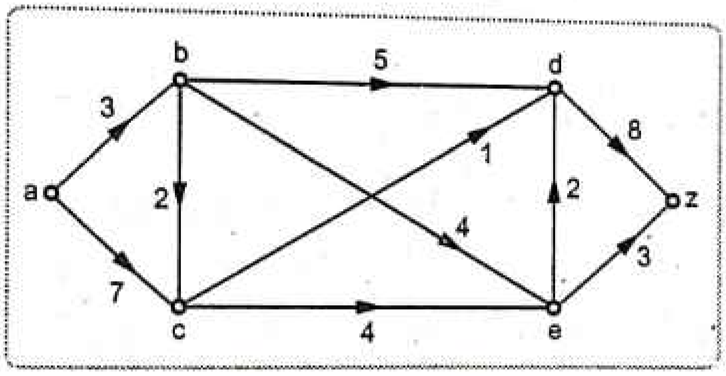
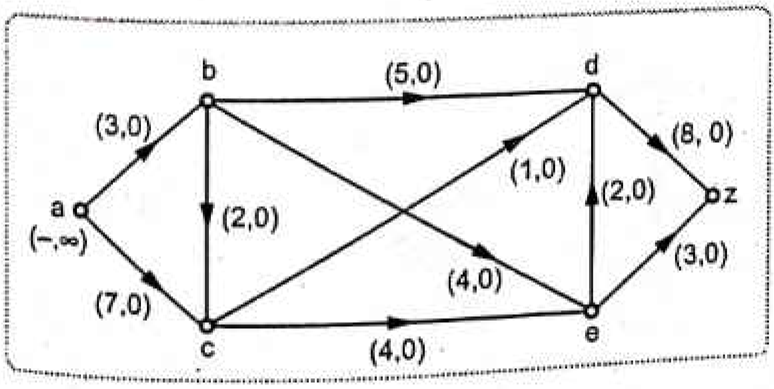



**Fig. Q.95.1**

**[SPPU : May-07, Dec.-14]**

**Ans. :** Step 1 : To find the maximum flow in the given transport network, assign the flow zero to each edge and label (–, ∞) to the source a.



**Step 2 :** Since the vertices b and c are adjacent to the source a, therefore label the vertices b and c as $(a^+, \Delta b)$ and $(a^+, \Delta c)$ respectively.

Where $\Delta b = ((a, b) - f(a, b) = 3$

$\Delta c = ((a, b) - f(a, c) = 7$

$\therefore$ Label of b is $(a^+, 3)$ and label of c is $(a^+, 7)$

Now the vertices d and e are adjacent to labeled vertices b $(a^+, 3)$ and $(a^-, 7)$ therefore we label the vertices d and e as $(b^+, \cdot d)$ and $(c^+, \cdot e)$ where

$$\Delta d = \min(\Delta b, c(b, d) - f(b, d))$$
$$= \min(3, 5, -0)$$
$$= 3$$

Similarly $\Delta e = \min(\Delta c, c(c, e) - f(c, e))$

$$= \min(7, 4, -0)$$
$$= 4$$

$\therefore$ Label of d is $(b^+, 3)$ and label of e is $(c^+, 4)$

Now the sink z is adjacent to both the vertices d and e, we can choose any vertex d or e to label the sink z.

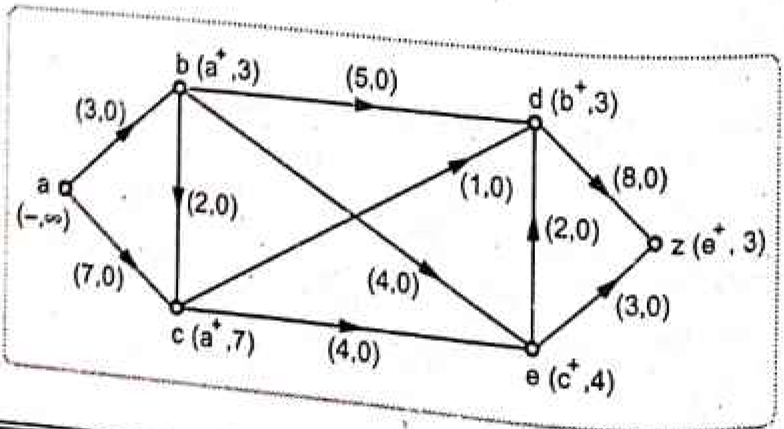
Let us choose the vertex e, which is labeled as $(c^+, 4)$

The label of z will be $(e^+, \Delta z)$ where

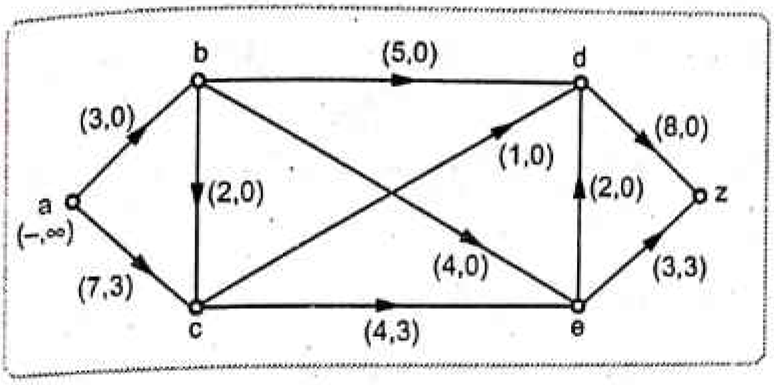
$$\Delta z = \min(4, 3, -0)$$
$$= 3$$

Hence label of z is $(e^+, 3)$

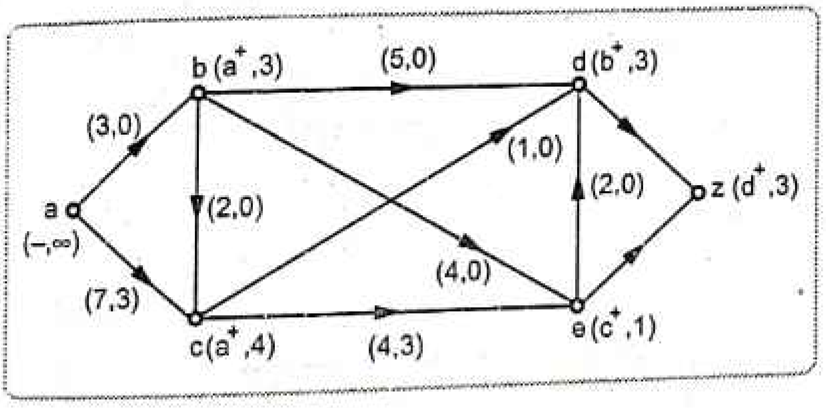
The labels of vertices can be shown as follows :



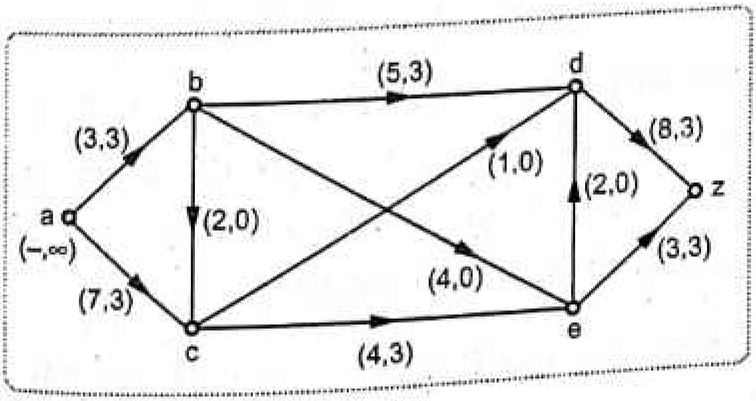
According to the label of the sink z, adjust the flow in the edges (e, z), (c, e) and (a, c)



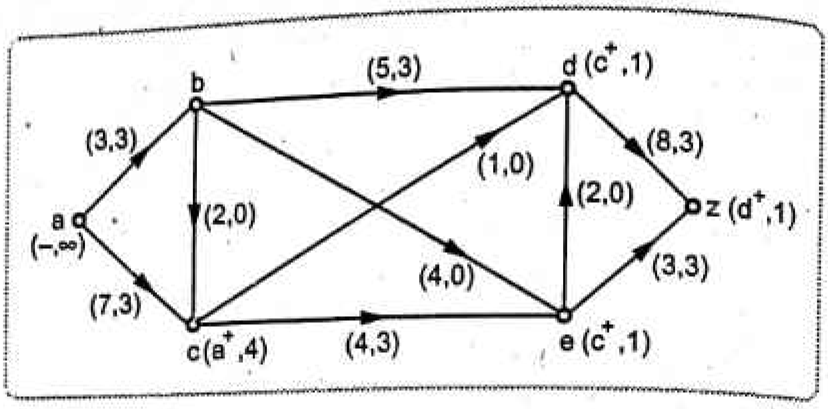

Repeat the step 2 in each pass, the new value of the flow is obtained.



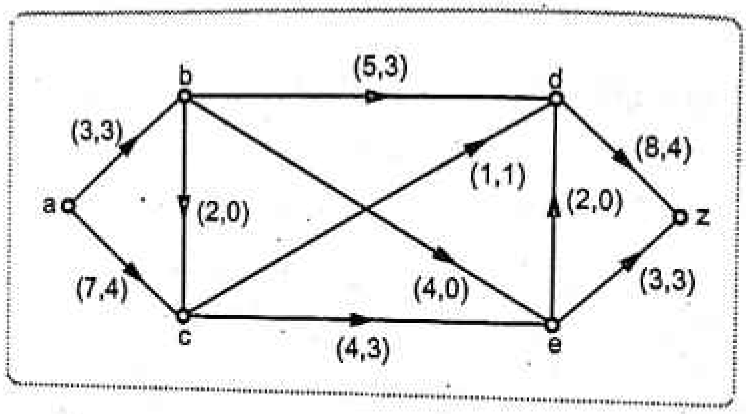

After adjusting the flow we have

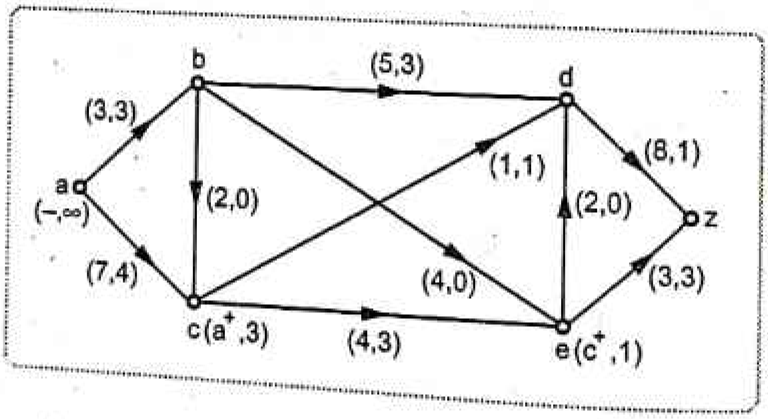


**Step 4 :**




Adjust the flow according to sink z label.




Since the vertex z (sink) cannot be labelled, we have to stop. The vertices b, d and z cannot be labelled because either the edges approaching to vertices are saturated or the edges in the opposite direction have flow zero. At this stage, we have minimum cut $(P, \bar{P})$.

P is the set of labelled vertices P = {a, c, e}. $\bar{P}$ is the set of unlabeled vertices, $\bar{P}$ = {b, d, z}

The edges in cut $P\bar{P}\{(a, b), (c, d), (e, z)\}$ capacity of the cut is $c(a, b) + c(c, d) + c(e, z) = 3 + 1 + 3 = 7$.

Hence the maximum flow in the given transport network 7.

**Q.96 Explain game tree.**

**Ans. :** A directed graph whose nodes are positions in a game and edges are moves, is called a game tree.

The complete game tree for a game is the game tree starting at the initial position and containing all possible moves from each position. The complete tree is the same tree as that obtained from the extensive form game representation.

We look at using trees for game planning, in particular the problems of searching game trees. We classify all games into following three types.

**1) Single player path finding problems :**

**For examples :**

a) Travelling salesman problem

b) Sliding puzzle

c) Hamiltonian paths

d) Rubik's cude

**2) Two player games :**

**For examples :**

a) Chess

b) Badminton

c) Tennis

d) Checkers

**3) Constraint satisfication problems :**

**For examples :**

a) Sudoku

b) Eight queen problem

c) Mathematical puzzles

d) Four queen problem

Each game consists of a problem space, an initial state and a set of goal states.

A problem space is a mathematical abstraction in a form of a tree :

- The root represents current state
- Nodes represent states of the game
- Edges represents moves
- Leaves represent final states (Win, loss or draw)

For example, in the 8-puzzle game

- **Nodes :** The different permentations of the tiles.
- **Edges :** Moving the black file up, down, right or left.

**Minimax :**

We consider game with two players in which one players gains are the result of another players losses so called zero sum games.

The minimax algorithm is a specialized search algorithm which returns the optimal sequence of moves for a player in an zero sum game.

In the game tree that result from the algorithm, each level represents a move by either of two players, say A and B player. The minimax algorithm explores the entire game tree using a depth-first search.

At each node in the tree where A-player has to move, A-player would like to play the move that miximizes the payoff. Thus, A-player will assign the maximum score amongst the children to the node where max makes a move. Similarly, B-player will minimize the pay off to A-player. The maximum and minimum scores are taken at alternating levels of the tree, Since A and B alternate turns. The minimax algorithm computes the minimx decision for the leaves of the game tree and than backs up through the tree to give the final value to the current state.

*END...*