Sol. (*i*) The binary tree after deleting node V is shown in Fig. 12.41.

(*ii*) The binary tree after deleting node E is shown in Fig. 12.42.

(*iii*) The binary tree after deleting root node R is shown in Fig. 12.43.

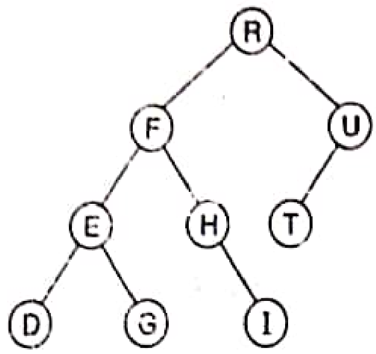


Fig. 12.41

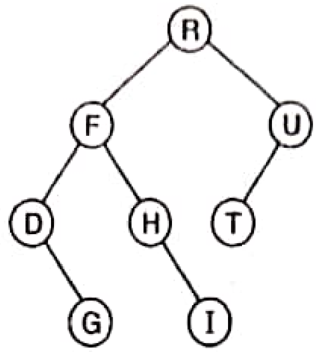


Fig. 12.42

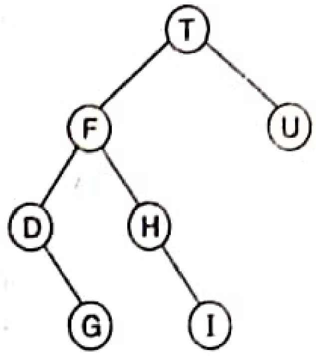


Fig. 12.43

12.17. SPANNING TREE

Consider a connected graph G = (V, E). A spanning tree T is defined as a subgraph of G if T is a tree and T includes all the vertices of G.

Example 27. *Draw all the spanning trees of the graph G shown in Fig. 12.44.*

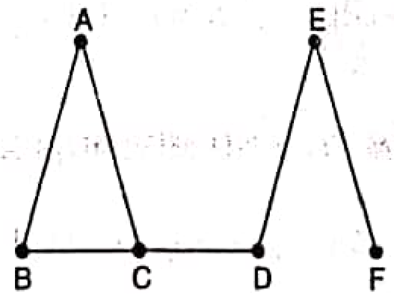


Fig. 12.44. Graph G.

Sol. All the spanning trees of graph G is as shown in Fig. 12.45.

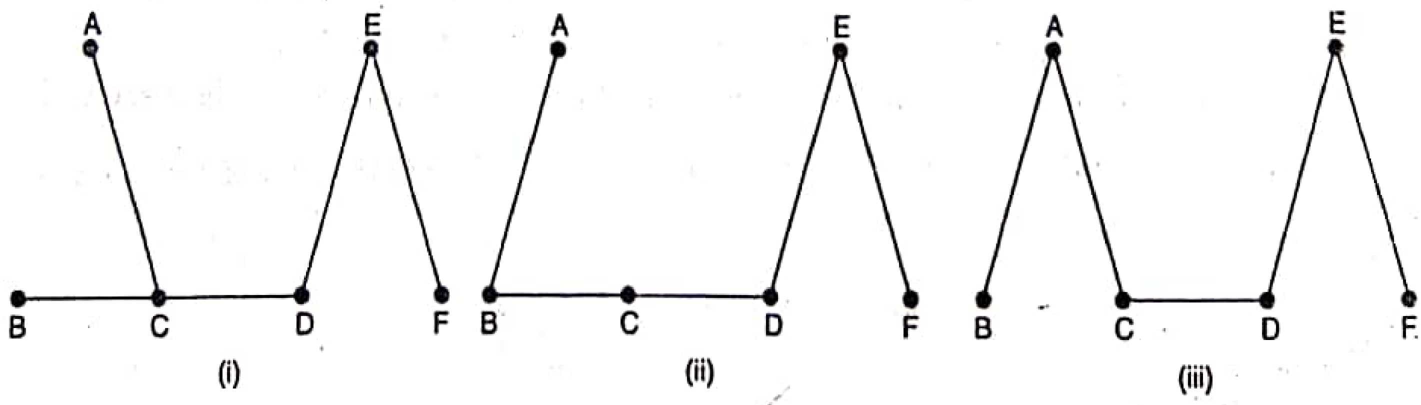


Fig. 12.45

12.18. APPLICATIONS OF TREES

12.18.1. Minimum Spanning Tree

Consider a connected weighted graph G = (V, E). A minimal spanning tree T of the graph G is a tree whose total weight is smallest among all the spanning trees of the graph G. The total weight of the spanning tree is the sum of the weights of the edges of the spanning trees.

The minimum weight of the spanning tree is unique but the spanning tree may not be unique because more than one spanning tree are possible when more than one edges exist having the same weight.

Theorem IV. *Prove that a simple graph is connected iff it has a spanning tree.*

(P.T.U. B.Tech. Dec. 2008)

Proof. First of all, suppose that a simple graph G has a spanning tree T. The tree T contains every vertex of G. Further, there is a path in T between any two of its vertices. Since T is subgraph of G, there is a path in G between any two of its vertices. Hence, G is connected.

Now, suppose that G is connected. If G is not a tree, then it must contain a simple circuit. Remove an edge from one of these simple circuits, the resulting subgraph has one fewer edge but still contains all the vertices of G and is connected. If this subgraph is not a tree, it has a simple circuit, so again remove an edge that is in simple circuit. Repeat this process untill no simple circuits remain. This is possible because there are only a finite number of edges in the graph. The process terminates when no simple circuits remain. A tree is produced since the graph is still connected as edges are removed. This is a spanning tree since it contains every vertex of G. Hence the theorem.

12.19. KRUSKAL'S ALGORITHM TO FIND MINIMUM SPANNING TREE

This algorithm finds the minimum spanning tree T of the given connected weighted graph G.

1. Input the given connected weighted graph G with n vertices whose minimum spanning tree T, we want to find.
2. Order all the edges of the graph G according to increasing weights.
3. Initialise T with all vertices but do not include any edge.
4. Add each of the graph G in T which does not form a cycle until $n - 1$ edges are added.

Example 28. *Determine the minimum spanning tree of the weighted graph shown in Fig. 12.46.*

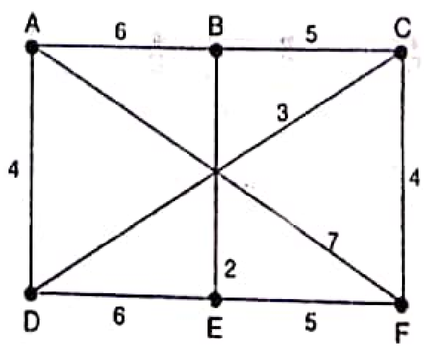


Fig. 12.46

Sol. Using Kruskal's algorithm, arrange all the edges of the weighted graph in increasing order and initialise spanning tree T with all the six vertices of G. Now start adding the edges of G in T which do not form a cycle and having minimum weights until five edges are not added as there are six vertices. (Fig. 12.47).

Edges	Weights	Added or Not
(B, E)	2	Added
(C, D)	3	Added
(A, D)	4	Added
(C, F)	4	Added
(B, C)	5	Added
(E, F)	5	Not added
(A, B)	6	Not added
(D, E)	6	Not added
(A, F)	7	Not added.

Minimum Spanning Tree

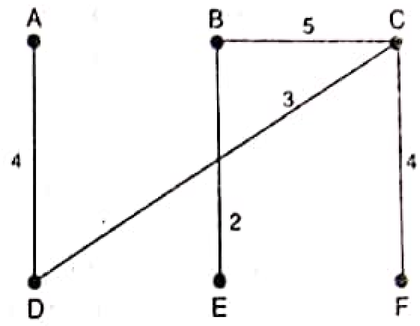


Fig. 12.47

Example 29. *Write a shortnote on* ***Prim's*** *and* ***Kruskal's*** *algorithms and execute them by giving a suitable example.* (P.T.U. B.Tech. May 2008)

Sol. Prim's Algorithm. Let R be a symmetric and connected relation with n vertices. The **Prime's algorithm** involves the following steps.

Step I. Choose a vertex v_1 of R. Let V = $\{v_1\}$ and E = { }

Step II. Choose a nearest neighbour v_i of V which is adjacent to v_j where $v_i, v_j \in$ V and for which the edge (v_i, v_j) does not form a cycle with members of E. Add v_i to V and (v_i, v_j) to E.

Step III. Repeat the step II until we get E = $n - 1$

Then V contains all n vertices of R and E contains the edge of a minimum spanning tree for R.

Example 30. *We find the minimal spanning tree for the graph R shown below (Fig. 12.48).*

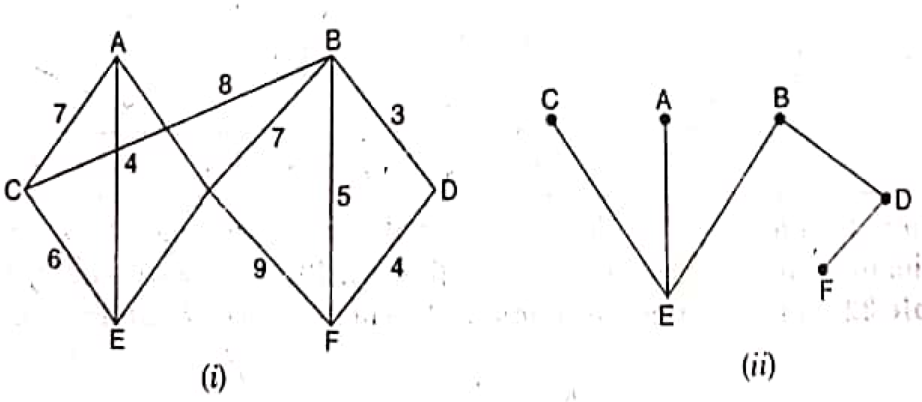


Fig. 12.48

Sol. This graph R [Fig. 12.48(*i*)] has 6 vertices, namely, A, B, C, D, E, F. Therefore, any spanning tree of R will have 5 edges. By Prim's algorithm, the edges are ordered by decreasing lengths and are successively deleted (without disconnecting R) until we have five edges remain. This gives the following data.

Edges	AF	BC	AC	BE	CE	BF	AE	DF	BD
Length	9	8	7	7	6	5	4	4	3
Deleted edges	✓	✓	✓	×	×	✓	×	×	×

Hence the minimal spanning tree of R will contains the edges {BE, CE, AE, DF, BD}. This spaning tree has length 24 as shown in Fig. 12.48(*ii*).

Kruskal's algorithm. Let R be a symmetric and connected relation with n vertices and let $S = [e_1, e_2, \ldots, e_k]$ be the set of weighted edges of R.

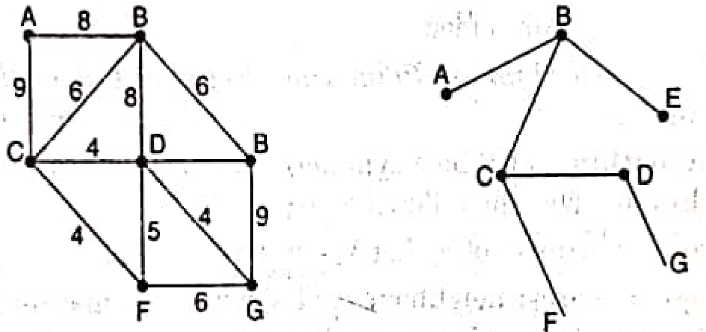
The Kruskal "algorithm involves the following steps".

Step I. Choose an edge e_1 in S of least weight. Let $E = \{e_i\}$. Replace S with $S - \{e\}$

Step II. Select an edge in S of least weight that will not make a cycle with members of $\{e_i\}$ and S with $S - \{e_i\}$.

Step III. Repeat step II until we get $E = n - 1$

Example 31. *Consider the graph R as shown below (Fig. 12.49). We find the minimal spanning tree of R.*



(*i*) (*ii*)

Fig. 12.49

Sol. The graph R [Fig. 12.49(*i*)] has 7 vertices namely A, B, C, D, E, F and G. Therefore, any spanning tree of R will have 6 edges. By Kruskal is algorithm, the edges are odered by increasing length and are successively **added** (without forming any cycle) until 6 edges are included. This gives the following data.

Edges	CD	CF	DG	DF	BC	BE	FG	DE	AB	BD	AC	EG
Length	4	4	4	5	6	6	6	7	8	8	9	9
Added edges	✓	✓	✓	×	✓	✓	×	×	✓	×	×	×

Therefore, the minimal spanning tree of R will contain the edges {CD, CF, DG, BC, BE, AB}. The minimum spanning tree of the graph [Fig. 12.49(*i*)] is shown in Fig. 12.49(*ii*).

Example 32. *Find a minimum spanning tree of the labelled connected graph shown in Fig. 12.50.*

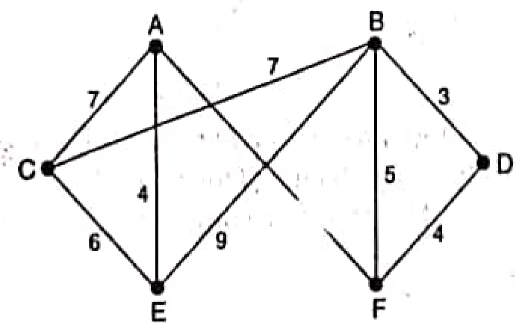


Fig. 12.50

Sol. Using KRUSKAL'S ALGORITHM, arrange all the edges of the graph in increasing order and initialize spanning tree with all the vertices of G. Now, add the edges of G in T which do not form a cycle and have minimum weight until $n-1$ edges are not added, where n is the number of vertices. The spanning tree is shown in Fig. 12.51.

Edges	Weights	Added or Not
(B, D)	3	Added
(A, E)	4	Added
(D, F)	4	Added
(B, F)	5	Not added
(C, E)	6	Added
(A, C)	7	Not added
(B, C)	7	Added
(A, F)	8	Not added
(E, B)	9	Not added

Minimum Spanning Tree

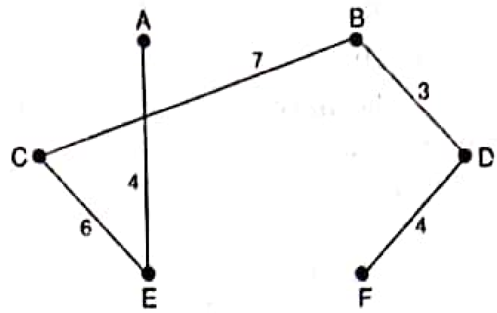


Fig. 12.51

The minimum weight of spanning tree is = 24.

Example 33. *Find all the spanning trees of graph G and find which is the minimal spanning tree of G shown in Fig. 12.52.*

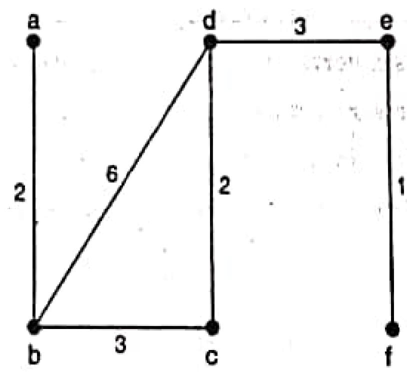
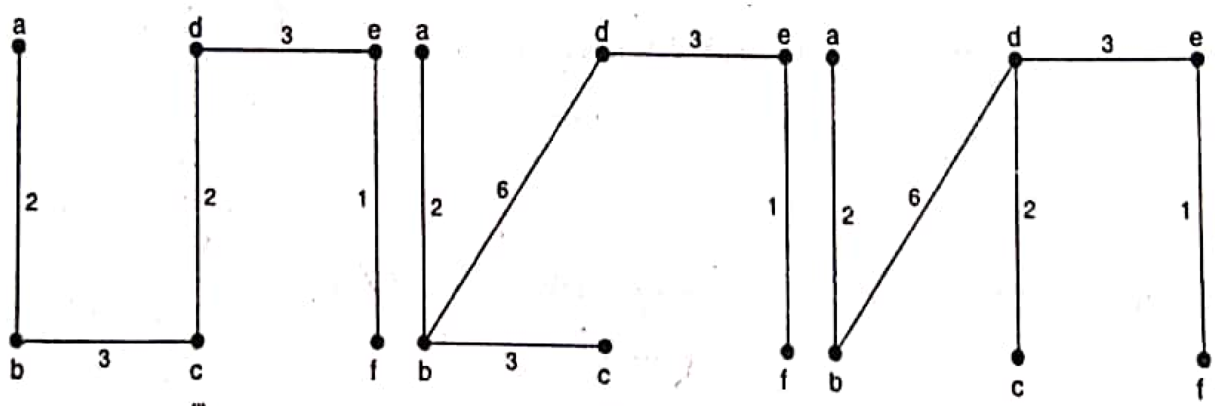


Fig. 12.52

Sol. There are total three spanning trees of the graph G which are as shown in Fig. 12.53.



(i) (ii) (iii)

Fig. 12.53

To find the minimal spanning tree, use the KRUSKAL'S ALGORITHM. The minimal spanning tree is shown in Fig. 12.54.

Edges	Weights	Added or Not
(E, F)	1	Added
(A, B)	2	Added
(C, D)	2	Added
(B, C)	3	Added
(D, E)	3	Added
(B, D)	6	Not added.

Minimal Spanning Tree

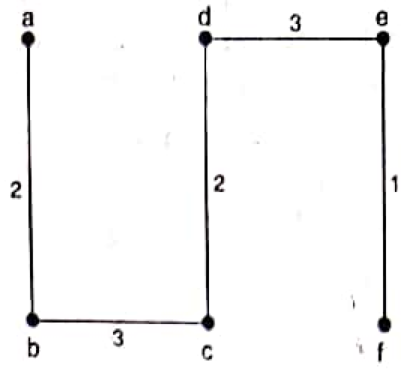


Fig. 12.54

The first one is the minimal spanning having the minimum weight = 11.

Example 34. *What are the properties of minimum spanning tree.*

Sol. Properties of Minimum spanning tree

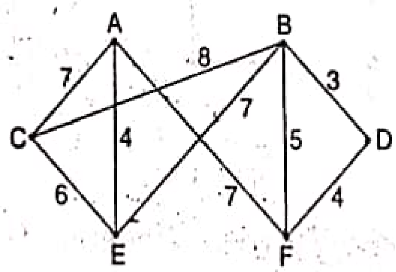
A minimum spanning tree T of a graph G is a tree whose total weight is the smallest among all the spanning trees of the graph G. It has the following properties.

(*i*) The total weight of the spanning tree is the sum of the weights of the edges of the spanning trees.

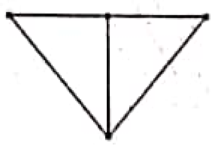
(*ii*) The minimum weight of the spanning tree is unique.

TEST YOUR KNOWLEDGE

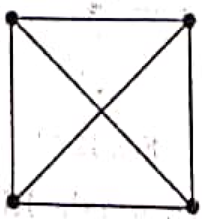
1. Draw all trees with exactly six vertices. *(P.T.U. B.Tech. Dec. 2013)*
2. Draw all trees with five or fewer vertices.
3. Find the number of trees with seven vertices.
4. Find a minimum spanning tree of the weighted graph shown below :



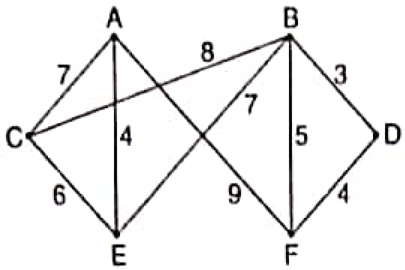
5. Find all spanning trees of the graph shown below :

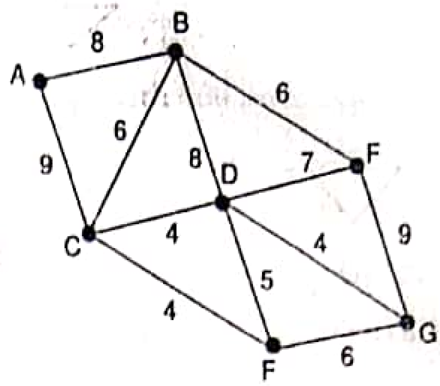


6. Find all spanning trees of the graph shown in the following figure.

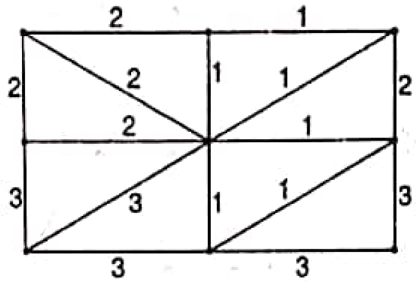


7. Find the minimal spanning tree of the following graph

(a) 

(b) 

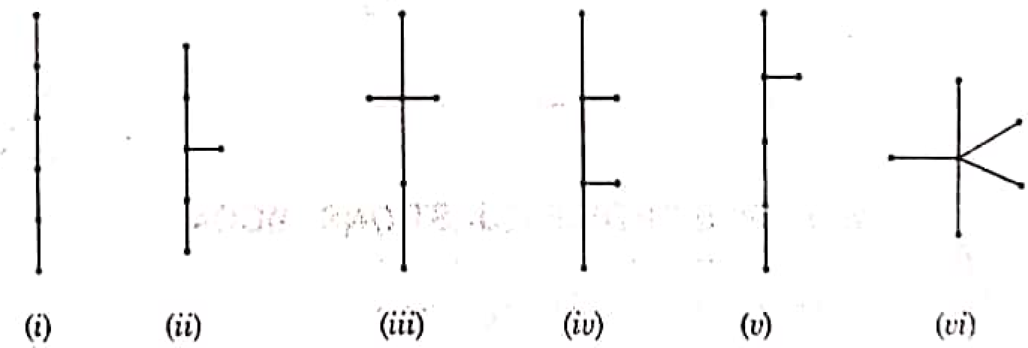
(c) Find the minimal spanning tree T for the weighted graph shown below :



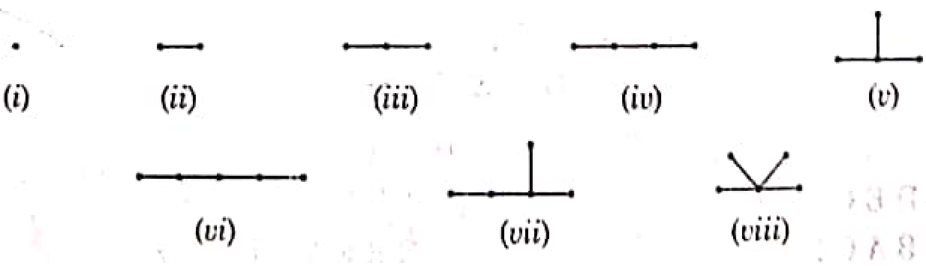
8. Show that the sum of the degrees of the vertices of a tree with n vertices is $2n - 2$.

Answers

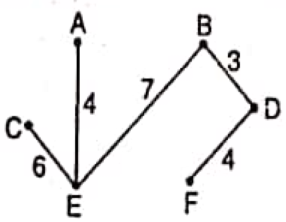
1. There are six such trees shown below :



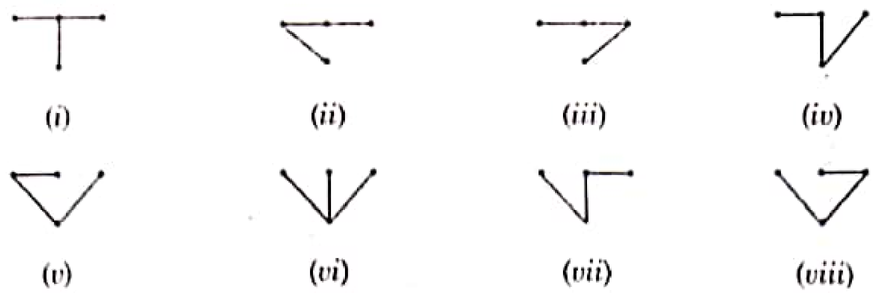
2. There are eight such trees shown below :



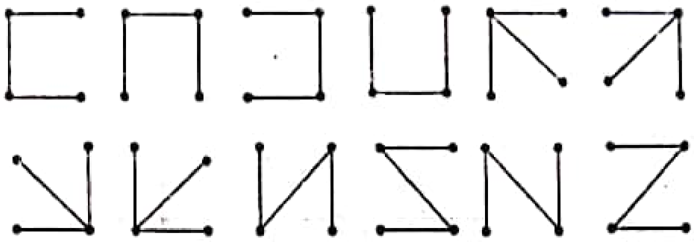
3. 15

4. 

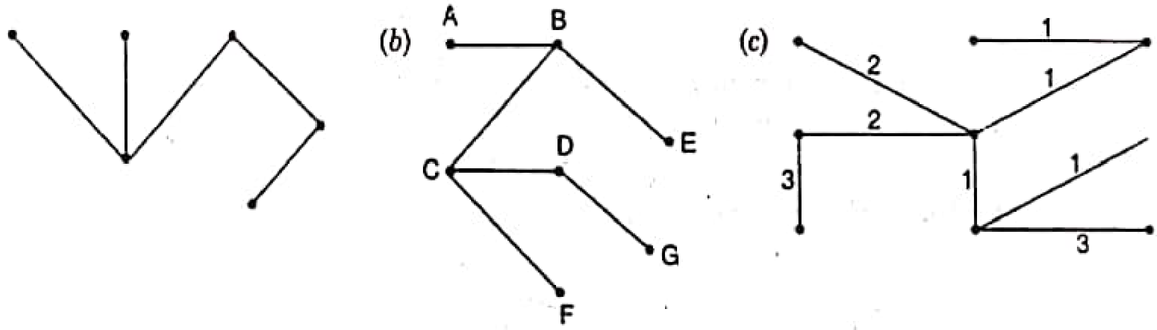
5. There are eight such spanning trees shown below :



6. There are twelve such spanning trees shown below.

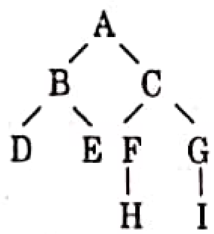


7. (a)



MULTIPLE CHOICE QUESTIONS (MCQs)

1. For the given binary tree, the preorder traversal is



(a) A B D E C F H G I
(b) A B C D E F G H I
(c) D E B A C F G H I
(d) D B E A H F C I G.

2. Let T be a full binary tree with I internal nodes. Then which of the following statements is TRUE?
(a) T has $2i + 1$ total nodes and $i + 1$ terminal nodes.
(b) T has $2i$ total nodes and $i + 1$ terminal nodes.
(c) T has $2i + 2$ total nodes and $i + 2$ terminal nodes.
(d) T has $2i^2$ total nodes and i^2 terminal nodes.

3. The maximum number of nodes in a binary tree of depth d is
   (a) $2d - 1$ (b) $2^d - 1$
   (c) $2d + 1$ (d) $2^d + 1$.
4. The number of external nodes in a full binary tree with 500 internal nodes are
   (a) 501 (b) 1000
   (c) 500 (d) Any number.
5. The total number of edges in any tree with n vertices is
   (a) $n(n-1)/2$ (b) $n/2$
   (c) n (d) $n - 1$.
6. Suppose T is a binary tree with 20 nodes. What is the minimum possible depth of T?
   (a) 1 (b) 3
   (c) 4 (d) 5.
7. If the height of a tree is 15, the highest level of the tree is
   (a) 15 (b) 14
   (c) 3 (d) 5.
8. In a postorder traversal, the _____ is processed first
   (a) Left subtree (b) Right subtree
   (c) Root (d) Any of the three.
9. Which of the following traversal techniques lists the nodes of binary search tree in ascending order?
   (a) Preorder (b) Postorder
   (c) Inorder (d) Level order.
10. Which traversal of the BST would print result in original order of input?
    (a) Preorder (b) Postorder
    (c) Inorder (d) Level order.

Answers and Explanations

1. (a) Apply the algorithm and check.
2. (a) External nodes in a tree are one more than internal nodes.
3. (b) The maximum number of nodes in a binary tree at depth d is one less than 2^d.
4. (a) The number of external nodes is one more than number of internal nodes.
5. (d) The number of edges needed to connect vertices of a tree is one less than the number of vertices.
6. (d) Draw the tree and see.
7. (b) The level of the tree is one less than its height.
8. (a) First the left subtree, then right subtree and then root.
9. (c) Inorder traversal of BST gives nodes in ascending order.
10. (c) Preorder traversal of BST gives nodes in original order.