8

TREES

A **connected graph** that contains no circuit is called a **tree.** Trees were used as long ago as 1857, when the English mathematician **Arthur Cayley** used them to count certain types of chemical compounds. Since that time, trees have been employed to solve problems in a wide verity of disciplines.

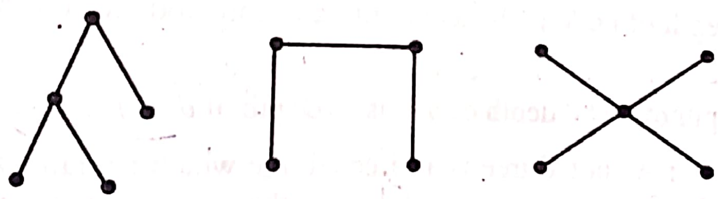
Trees are particularly useful in computer science Trees are employed to construct efficient algorithms for locating items in a list. They are used to construct networks with the least expensive set of telephone lines linking distributed computers. Trees can be used to construct efficient codes for storing and transmitting data. Trees can model procedures that are carried out using a sequence of decisions. This makes trees valuable in study of Computer Science.

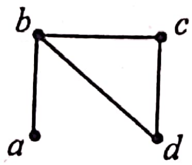
Definition Tree : A graph G is called a tree if

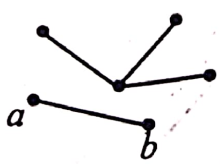
(*i*) G is connected

(*ii*) G has no cycles.

Following graphs are trees :



Graph  is not tree as it contains a cycle *b*, *c*, *d*, *b*

Similarly  is not tree as it contains a disconnected component *a b*.

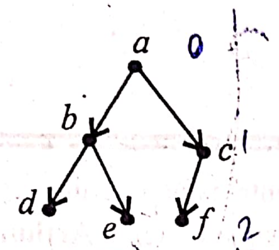
Remark : From above definition, it follows that a tree has to be a simple graph *i.e.* having neither a self loop nor parallel edges because both of them from cycle.

Note : Tree is said to be directed if every edge of tree is assigned a direction, otherwise tree is undirected.

TERMINOLOGY USED IN TREE

When we discuss tree, we encounter a number of terms that are necessary to understand.

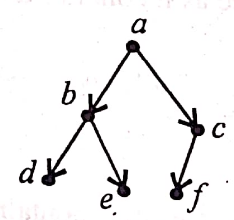
The terms used are :



- **NODE OR VERTEX:** Node is key component of tree which stores information and can have one or more links for connecting to other nodes.
- **EDGE or LINK:** A directed line from one node to other node is called edge, link, arc or branch of a tree. In above figure ab, ac etc are links.
- **ROOT:** The vertex having indegree zero is called root of tree. In above tree root of tree is a.
- **PATH :** A Path is a sequence of nodes when we traverse from one node to other along the edges which connect them e.g. path from a to f is a, c, f.
- **LEVEL:** Level of node is an integer value that measures the distance of a node from the root. Root is at level 0. The child(s) of root are at level 1 and so on.
- **HEIGHT:** Height of node is the length of longest path from node to a leaf. All the leaves are at height 0. Height of root is height of tree. In above tree, height of d, e, f is 0, Height of b, c is 1 and height of a is 2.
- **DEPTH:** The depth of node is the length of path from node to root of tree Root has depth 0.

  In above tree depth of a is 0 depth of b, c is 1. Depth of d, e, f is 2.
- **ROOTED TREE :** A rooted tree is a directed tree which contains a unique vertex 'r' such that in-degree of r is zero and every other vertex has in-degree one. The vertex 'r' is called root of rooted tree.

  For example :



  is a rooted tree with root 'a'.
- **PARENT AND OFFSPRING :** If (x, y) is any directed edge then x is called parent of y and y is called offspring of x. Root of tree has no parent whereas every other node has a unique parent. A parent can have several offsprings. Offspring is also called **child** or **son.**

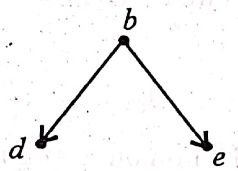
  In above tree a is parent of b and c. b has two offsprings d and e.
- **LEAF :** A node having no offsprings (outdegree = 0) is called a leaf. In fig. d, e, f are leaves. Leaf is also called **external** or **terminal** node.

- **SIBLINGS :** Two nodes having same parent are called siblings. In figure b, c are siblings of a.
- **INTERIOR NODE :** Node having at least one child is called Interior node.
- **ANCESTOR :** Ancestors of a vertex other than root are the vertices in the path from root to this vertex, excluding the vertex itself and including the root. For example, ancestors of d are b and a.
- **DESCENDANT :** Descendants of a vertex 'V' are those vertices that have 'V' as an ancestor.

For example : Descendants of b are d and e.

- **SUBTREE :** If 'a' is any vertex in a tree, the subtree with a as its root is the subgraph of tree consisting of a and its descendants are all edges incident to these descendants.

In given fig. subtree of b is T(b) as shown :

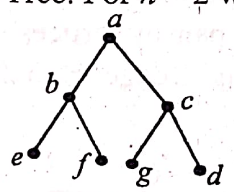


- **FOREST:** A forest is an undirected graph whose components are all trees.
- **BINARY TREE :** (P.T.U. B.C.A.-I 2007)

Let T is a tree. We say T is n-tree or n-ary tree if every vertex has at most n offsprings. In particular, if $n = 2$ then tree is called binary tree. So binary tree is that tree in which every node can have 0, 1 or 2 offsprings.

- **COMPLETE BINARY TREE :** (P.T.U. B.C.A.-I 2006)

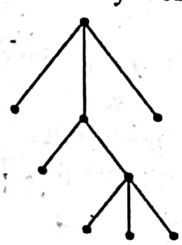
In n-tree, if every vertex of T, other than leaves, has exactly n-offsprings then we say T is complete n-Tree. For $n = 2$ we say tree is complete Binary tree.

Example (*i*) 

is a Binary Tree.

In this example, b is left child of a and c is right child of a.

Moreover every vertex (except leaves) has 2 children so tree is complete Binary Tree.

(*ii*) 
is a 3-ary tree. But is not complete.

Art-1. Properties of Tree

Property 1. There is one and only one path between every pair of vertices in a tree T.

(P.T.U. B.C.A.-I 2007)

Proof : Since T is a connected graph. Therefore there exists atleast one path between every pair of vertices in tree T.

Suppose that between two vertices v_1 and v_2 there exists two distinct paths. The union of these two paths will contain a circuit and then T cannot be a tree. Thus there is one and only path between every pair of vertices in a tree T.

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

(P.T.U. B.C.A.-I 2007)

Proof : Since there is one and only one path between every pair of vertices in G implies that G is connected graph. Suppose that G contain a circuit, then there is atleast one pair of vertices v_1, v_2 (say) such that there are two distinct path between them. A contradiction to the given fact and so G cannot have circuit. Hence G is a connect graph without circuit implies that G is a tree.

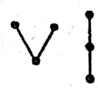
Property 3. A tree with n vertices has n–1 edges.

Proof. We shall prove the result by induction on the number of vertices n.

Obviously the result is true for n =1, 2, 3 as

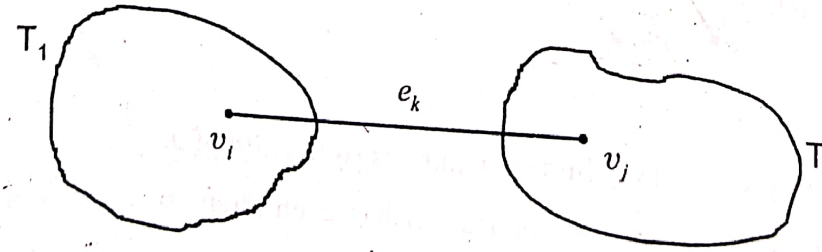
When $n = 1$ we have Zero edge

When $n = 2$ we have  One edge

When $n = 3$ we have  Two edge.

Let us assume that the result is true for all tree with less than n vertices.

Consider a tree T with n vertices. Let e_k be an edge with end verities v_i and v_j. Since there is one and only one path between every pair of vertices *i.e.*, there is no other path between v_i and v_j except e_k. Therefore deletion of edge from T will disconnect the graph as shown below.



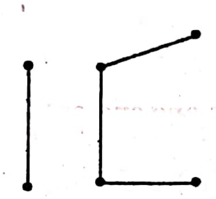
Therefore T− e_k consist of exactly two component T_1 and T_2 (say). Since there is no circuits in T each of these components is a tree. Further, both of these tree T_1 and T_2 have less then n–vertices, therefore by supposition, each tree will contain edge one less then the number of vertices in it. So T− e_k consist of $(n-2)$ edges implies that T has exactly $(n-2)+1 = n-1$ edges. This completes the induction.

Note : It may be noted that the vertices of a tree are connected together with the minimum number of edges.

Definition. Minimally connected graph :

A connected graph G is said to be minimally connected if removal of any edge from it disconnect the graph.

For Example :



are minimally connected graphs.

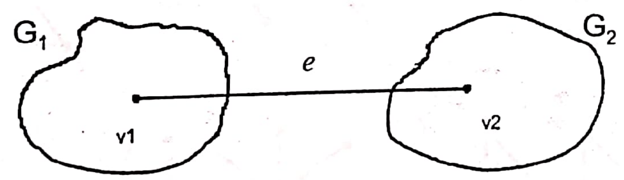
Property 4. A graph is a tree if and only if it is minimally connected.

Proof : Firstly, let the graph T be a tree. Therefore T must be a connected graph. If possible, let T be not minimally connected. Then there must exists an edge e_i in T such that $T - e_i$ is connected. Therefore e_i is in some circuit which implies that T is not a tree, a contradiction. Hence T must be minimally connected.

Conversely. Let T be a minimally connected graph. Therefore T cannot have a circuit otherwise we could remove one of the edge in the circuit and still leave the graph connected. Hence T is a tree.

Property 5. A graph G with n vertices and $(n - 1)$ edges and no circuit is connected.

Proof : Let there exist a graph G with n vertices, $(n-1)$ edges and no circuit which is disconnected. Then G will consists of two or more circuit-less component. Without loss of generality, let G consist of two components G_1 and G_2 as shown below :



Now, add an edge e between the vertices v_1 in G_1 and v_2 and G_2. Since there is no path between v_1 and v_2 in G so by adding an edge e did not create a circuit in G. Thus $G \cup e$ is a circuit less connected graph. *i.e.*, $G \cup e$ is a tree in other words, a tree has n vertices an n edges, which is not possible. Hence a graph with n vertices and $(n - 1)$ edges and no circuit is connected.

Remark : Five different but equivalent definition of tree are A graph G with n vertices is called a tree if

(*i*) G is connected and has no circuit.

(*ii*) G is connected and has $(n - 1)$ edges.

(*iii*) G has $n - 1$ edges and no circuit.

(*iv*) there is exactly one path between every pair of vertices in G.

(*v*) G is minimally connected.

Art-2. In any non-trivial tree, there are atleast two pendent vertices.

or

In any non-trivial tree, there are at least two vertices of degree 1.

Proof : Let T be a non–trivial tree with n vertices, then T has n–1 edges.

$\therefore$ By fundamental theorem on graph theory

$$\sum_{i=1}^{n} \deg (v_i) = 2(n-1) = 2n-2 \qquad \ldots(1)$$

If possible, let T contain only one vertex (say) v_1 of degree 1. Then

$$\deg (v_1) = 1 \quad \text{and} \quad \deg (v_i) \geq 2 \quad \text{for } i = 2, 3, 4, \ldots, n$$

$$\therefore \quad \sum_{i=1}^{n} \deg (v_i) = \deg (v_1) + \sum_{i=2}^{n} \deg (v_i)$$

$$= 1 + \sum_{i=2}^{n} \deg (v_i) \geq 1 + 2(n-1) = 2n-1 \qquad \ldots(2)$$

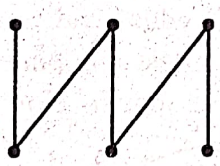
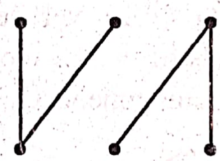
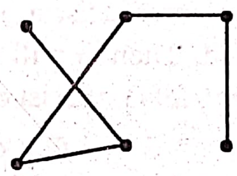
$\therefore$ From (1) and (2), we get

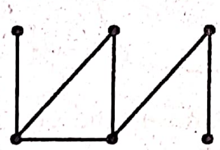
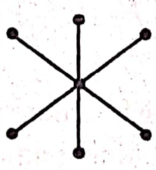
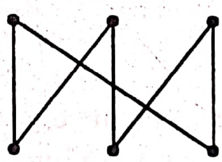
$2n - 2 \geq 2n - 1$, a contradiction.

$\therefore$ T must have more than one vertex of degree 1.

i.e., T has atleast two vertices of degree 1.

Example 1: Which of following graphs are trees ?

(*a*)  (*b*)  (*c*) 

(*d*)  (*e*)  (*f*) 

Sol. (*a*) Number of vertices = 6

Number of edges = 5

as edges = $n-1$ and graph has no cycle so it represent a tree.

(*b*) Since graph contains two disconnected components so it is not a tree.

(*c*) Number of vertices = 6

Number of edges = 5

as edges = $n-1$ and graph has no cycle so it represent a tree.

(*d*) Graph contains a cycle so it is not a tree.

(e) Graph is a tree ∵ it contains no cycle and number of edges (6) = number of vertices (7) − 1.

(f) Graph is not tree as it contains a cycle.

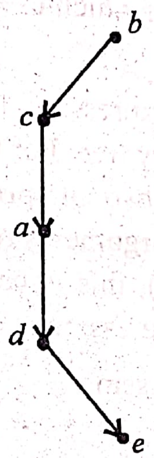
xample 2. Let A = {a, b, c, d, e} R = {(a, d), (b, c), (c, a), (d, e)} check whether R is a ee. If it is, find the root.

ol. We are given five vertices and four edges. R will from tree iff

(i) edges connect all vertices

(ii) edges will not form cycle among vertices.

From R, it is clear 'b' has indegree 0. If R is tree then b must be root. So taking b as ot we construct all edges as below



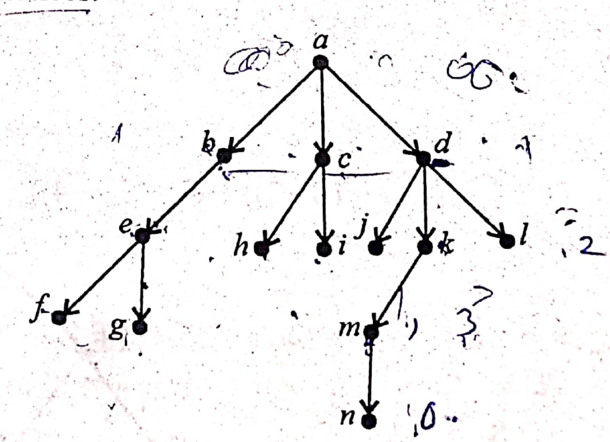
As no cycle is formed, so R is a tree and b is root of tree.

xample 3. Prove that A = {1, 2, 3, 4, 5, 6} R = {(1, 1), (2, 1), (2, 3), (3, 4), (4, 5), (4, 6)} not a tree.

ol. As number of vertices = 6 and number of edges = 6. So R cannot be a tree.

∵ in tree no. of edges = $n-1$, n = no. of vertices.

xample 4. Consider the tree.



(a) List all level-3 vertices.

(b) List all leaves.

(c) What are siblings of d?

(d) Draw the Tree T(b).

(e) What is level and height of m?

ol. (a) Root of tree is a. So level of a = 0

At level one we have b, c, d

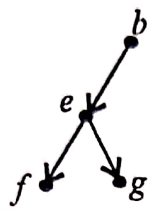
at level two we have e, h, i, j, k, l

at level three we have f, g and m.

(b) Leaves are vertices that have no offsprings. In given tree leaves are f, g, h, i, j, n nd l.

(c) Siblings of d are b and c $\because$ b, c, d have same parent a.

(d) Tree T(b) is as shown



(e) Level of $m = 3$

Height of $m = 1$

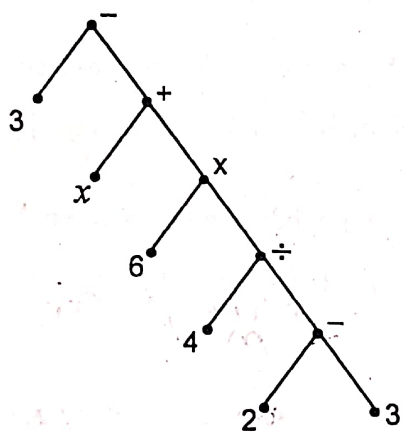
Art 3. Labeled Tree

Definition : A Tree is said to be labeled in which every vertex of Tree has assigned a unique label.

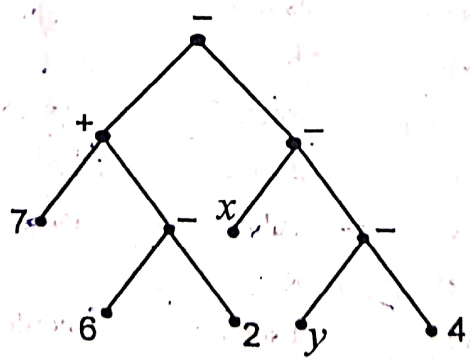
Labeled tree is usually used to construct expression Tree. Any algebraic expression can be represented with the help of labeled binary tree. For this root of tree is labeled with the central operator of main expression. The two offsprings of root are labeled with central operator of expression for left and right arguments respectively. If either argument is a constant or variable (instead of expression), this is used to label the corresponding offspring vertex. This process continues until expression is exhausted.

Example 5. Construct the tree of algebraic expression.

(i) $3 - (x + (6 \times (4 \div (2-3))))$



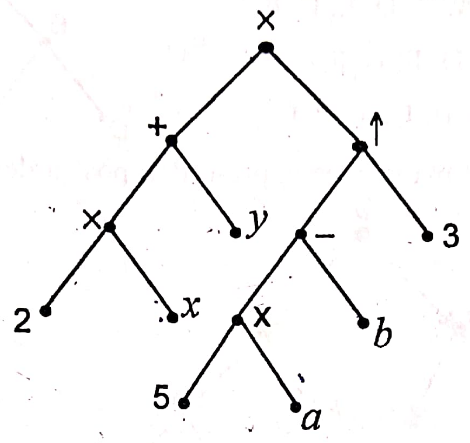
(ii) $(7 + (6-2)) - (x - (y-4))$



(iii) $(2x + y)(5a - b)^3$.

This expression is equivalent to $((2 \times x) + y) \times ((5 \times a) - b)^3$

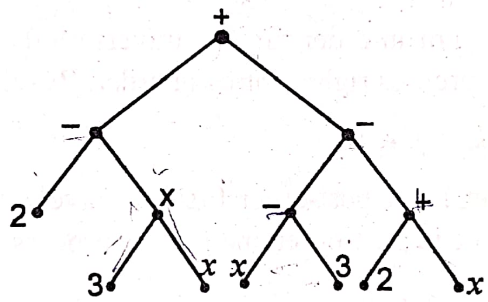
Exponent is represented by symbol ↑



Example 6. Draw a binary tree to represent $(2-(3\times x)) + ((x-3)-(2+x))$.

(Pbi. U. B.C.A. 2006)

Sol.



Art-4. Traversal of Binary Trees or Tree Searching (P.T.U. B.C.A.-I 2006) (Pbi. U. 2009)

Traversing means to visit each node of tree exactly once. There are three standard ways of traversing a binary tree T with Root R. These algorithms are called preorder, inorder and post order traversals and are as follows :

Preorder : (1) Process the root R.

(2) Traverse the left subtree of R in preorder.

(3) Traverse the right subtree of R in preorder.

Inorder : (1) Traverse the left subtree of R in inorder.

(2) Process the root R.

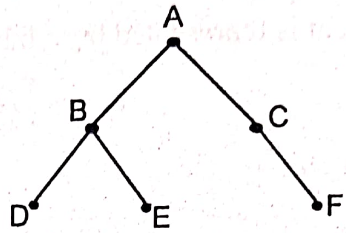
(3) Traverse the right subtree of R in inorder.

Post order : (1) Traverse the left subtree of R in postorder.

(2) Traverse the right subtree of R in postorder

(3) Process the root R.

Example 7. Traverse the following Tree in Preorder, Post order and inorder.

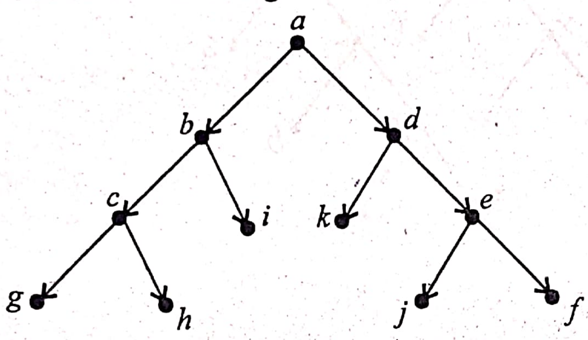


Preorder Traversal : A, B, D, E, C, F

Postorder Traversal : D, E, B, F, C, A

Inorder Traversal : D, B, E, A, C, F.

Example 8. Search the following Tree in pre-order, post-order and in-order.



(Pbi. U. B.C.A. April 2006)

Sol. Pre-order Searching : For pre-order first we process root then left subtree in preorder and then right subtree in pre-order. Result is :

$a, b, c, g, h, i, d, k, e, j, f$

In-order Searching : For in-order first we traverse left subtree in inorder, then we process root and atleast we process right subtree in order. Result is :

$g, c, h, b, i, a, k, d, j, e, f$

Post-order Searching : For post-order first we traverse left subtree in post-order, then we traverse right subtree in post-order and last we process root. Result is :

$g, h, c, i, b, k, j, f, e, d, a.$

Polish Notations : An expression tree has three forms

1. Prefix Form : When a pre-order traversal is performed on an expression tree then result obtained is called pre-fix form or Polish form of the given algebraic expression.

2. Post Fix Form : When a post-order traversal is performed on an expression tree then result obtained is called post-fix form or reverse polish form of the given algebraic expression.

3. Infix Form : Infix form results from the in-order traversal of expression tree.

Consider the expression $a + b$. In this expression a, b are operands and '+' is an operator. The sequence of operators and operands in three form is as given below :

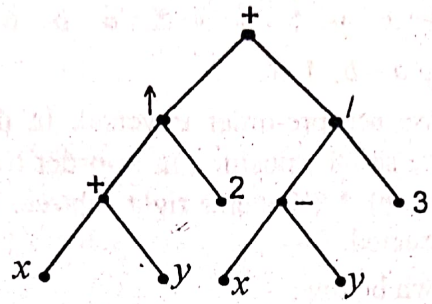
Pre-fix form : operator, operand, operand

Post-fix form : operand, operand, operator

In-fix form : operand, operator, operand.

Example 9. Find pre-fix and post-fix expression for $((x + y) \uparrow 2) + ((x - y)/3)$

Sol. The expression tree of above expression is show below :



To find pre-fix form of given expression, we traverse the tree in pre-order. Result is :

$+ \uparrow + x \ y \ 2 \ / - x \ y \ 3.$

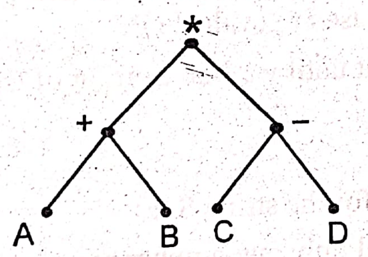
Post-fix form of expression is given by traversing the expression tree in post order. Result is :

$x \ y + 2 \uparrow x \ y - 3 \ / \ +.$

Example 10. Represent the expression (A + B) ∗ (C−D) as a binary tree and write prefix form of expression.

(Pbi. U. B.C.A. April 2006)

Sol. The Binary Tree corresponding to expression is shown below



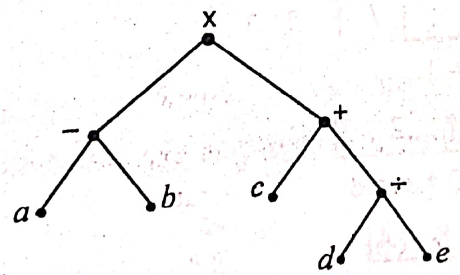
Prefix form : ∗, +, A, B, −, C, D.

Example 11. Consider the completely parenthesized algebraic expression :

$(a-b) \times (c + (d \div e))$. Find its preorder, post order and inorder search.

(B.C.A.-II, Sept. 2007)

Sol. First we draw expression tree corresponding to given algebraic expression which is shown below :



Preorder Search : $\times, -, a, b, +, c, \div, d, e.$

Postorder Search : $a, b, -, c, d, e, \div, +, \times$

Inorder Search : $a, -, b, \times, c, +, d, \div, e.$

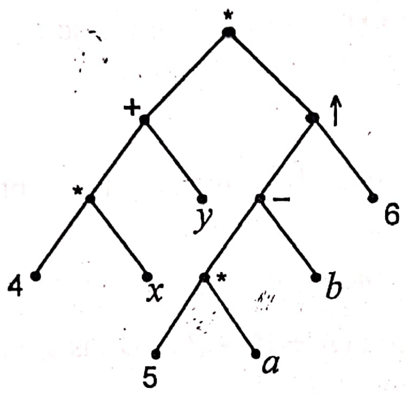
Example 12. Construct a tree whose pre-order and in-order traversal is given below :

Pre-order $* \; + \; * \; 4 \; x \; y \; \uparrow \; - \; * \; 5 \; a \; b \; 6$

In-order $(4\,x + y) * (5\,a - b) \uparrow 6$ (P.T.U. B.C.A.-I 2003)

Sol. To construct tree 1st we see pre-order traversal. In this first element is $*$ which become root of tree. Then we see the position in In-order traversal. Left of $*$ *i.e.* $4\,x + y$ become left subtree and $(5\,a - b) \uparrow 6$ become right subtree. This process will repeat until whole of tree has been constructed.

The tree formed is shown below :



Art-5. Evaluation of Prefix Expression (Polish Expression)

Procedure : Let P be given pre-fix expression. Let S denotes binary arithmetic operator $(+, -, *, /, \uparrow)$. Following steps are used to evaluate P :

1. Traverse P from left to right until we find a string of the form Sxy, where x and y are numbers (operands).
2. Evaluate xSy.
3. Substitute the result of xSy for the string Sxy.
4. Continue the procedure until only one number remains.

Example 13. Evaluate the pre-fix expression

$$+ - * 2 \; 3 \; 5 / \uparrow 2 \; 3 \; 4.$$

Sol. The step by step procedure to evaluate above expression is shown in following figure.

Step 1 : $+ \; - \underbrace{* \;\; 2 \;\; 3}_{2 * 3 = 6} \; 5 \; / \uparrow \; 2 \; 3 \; 4$

Step 2 : $+ \underbrace{- \;\; 6 \;\; 5}_{6 - 5 = 1} \; / \uparrow \; 2 \; 3 \; 4$

Step 3 : $+ \; 1 \; / \underbrace{\uparrow \;\; 2 \;\; 3}_{2 \uparrow 3 = 8} \; 4$

Step 4 : $+ \; 1 \underbrace{/ \;\; 8 \;\; 4}_{8/4 = 2}$

Step 5 : $\underbrace{+ \;\; 1 \;\; 2}_{1 + 2 = 3}$

Value of expression = 3.

Art-6. Evaluation of Post fix Expression (Reverse Polish Expression)

Procedure : Let P be given post-fix expression. Let S denotes binary arithmetic operator (+, −, *, /, ↑)

1. Traverse P from left to right until we find a string of the form $x\ y$ S, where x and y are numbers. (operands).
2. Evaluate xSy.
3. Replace string xyS by result of xSy.
4. Continue the procedure until only one number remains.

Example 14. What is the value of post-fix expression

7 2 3 * − 4 ↑ 9 3 / + (Pbi. U. B.C.A. 2012)

Sol. The step by step procedure to evaluate above expression is shown in following figure:

Step 1 : 7 $\underbrace{2\ 3\ *}_{2*3=6}$ − 4 ↑ 9 3 4 / +

Step 2 : $\underbrace{7\ 6\ -}_{7-6=1}$ 4 ↑ 9 3 / +

Step 3 : $\underbrace{1\ 4\ \uparrow}_{1\uparrow 4=1}$ 9 3 / +

Step 4 : 1 $\underbrace{9\ 3\ /}_{9/3=3}$ +

Step 5 : $\underbrace{1\ 3\ +}_{1+3=4}$

Value of expression = 4.

Example 15. Is there a binary tree with height 6 and 65 leaves ?

(Pbi. U. B.C.A. Sept. 2007)

Sol. Height of Tree = 6

We know, maximum number of leaves in a Binary Tree = 2^r, where r is height.

So maximum number of leaves = $2^6 = 64$

Therefore, binary tree with height 6 and 65 leaves is not possible.

Example 16. A tree with n vertices has at least two vertices of degree 1. ($n \geq 2$).

Sol. Given tree T has n vertices

So number of edges in T = $n - 1$

Let us suppose T has no vertex of degree = 1

We know, total degree $= 2 \times$ number of edges

$= 2\ (n - 1)$.

So degree of each vertex $= \frac{2n-2}{n} = 2 - \frac{2}{n} \leq 2 - \frac{1}{n}$ $\left(\because \frac{2}{n} \leq 1\right)$

$\Rightarrow$ either degree of vertex is 1 or 0.

But T can not have a vertex of degree zero.

So our supposition is wrong. T must have at least two vertices of degree = 1.

Example 17. How many edges does a tree with 10,000 vertices have ?

Sol. Number of vertices $= 10000$

We know number of edges $= n - 1$

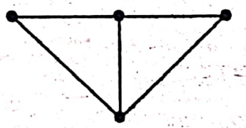
$= 10,000 - 1 = 9999.$

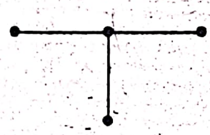
SPANNING TREE : (P.T.U. B.C.A.-I 2007)

Let G be a connected graph. A subgraph T of G is called a spanning tree if

(*i*) T is a tree.

(*ii*) T contains all vertices of G.

For example  is a connected graph G. Clearly it is not a tree.

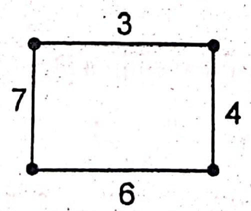
A spanning tree of G is 

Note : Spanning tree of graph is not unique.

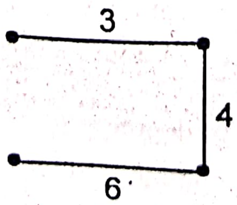
MINIMAL SPANNING TREE : (P.T.U. B.C.A.-I 2006)

A minimal spanning tree of a weighted graph is a spanning tree with the condition that sum of weights of tree is as small as possible.

For example : Given weighted graph



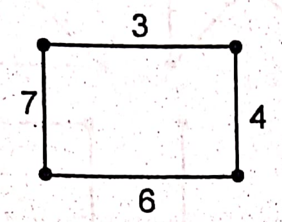
Minimal spanning tree of this graph is



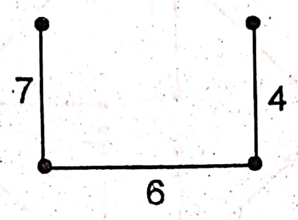
Weight of minimal spanning Tree $= 3 + 4 + 6 = 13.$

Maximal Spanning Tree : A maximal spanning tree of a weighted graph is a spanning tree with the condition that sum of weights of tree is as large as possible.

For example : Given weighted graph



Maximal spanning tree of this graph is



Weight of maximal spanning tree $= 7 + 6 + 4 = 17$.

Art-7. A graph G is connected if and only if it has a spanning tree.

(Pbi. U. B.C.A.-II 2011)

Proof : Firstly, let the graph G be connected. Let k be the number of circuits (or cycles) in G. We apply induction on k. If $k = 0$, then G has no circuit also, G is connected.

$\Rightarrow$ G is a tree and so has a spanning tree.

$\therefore$ The result is true for $k = 0$

Let the result is true for all connected graphs with less than k cycles.

Let G be a connected graph with k circuits. Let e be an edge in one of the circuits, then $G-e$ is connected graph having fewer edges than G.

$\therefore$ By induction hypothesis $G - e$ has a spanning tree. But $G - e$ has all the vertices of G.

$\therefore$ The spanning tree of $G - e$ is also a spanning tree for G.

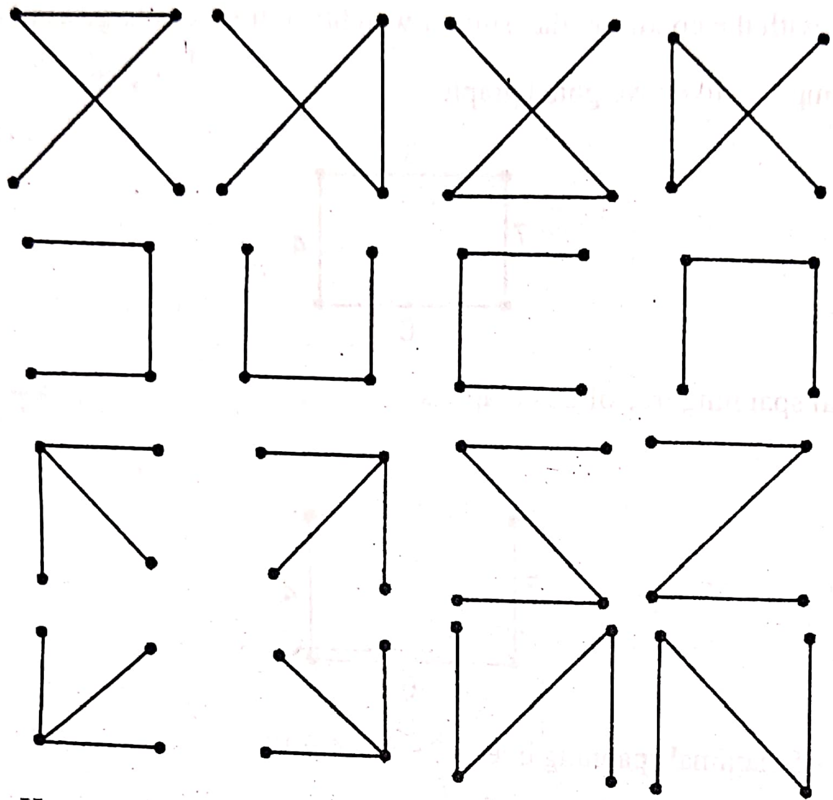
$\therefore$ The result is true for G also. Hence the result follows by induction.

Conversely : Let the graph G has a spanning tree T (say). We show that G is connected. Since T is a spanning tree of G. Therefore there exists a path between any pair of vertices in G along the tree T.

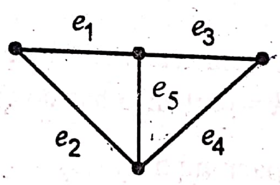
$\therefore$ G is connected.

Remark. (Cayley's Theorem) The complete graph K_n has n^{n-2} different spanning tree.

For example : There are 16 spanning trees of K_4. These are as shown below :



Example 18. How many spanning trees the graph have ? Draw all spanning trees of graph.



Sol. We know spanning tree contains all vertices of graph. So number of edges taken $= 4 - 1 = 3$.

As we are given 5 edges which means we have to remove 2 edges.

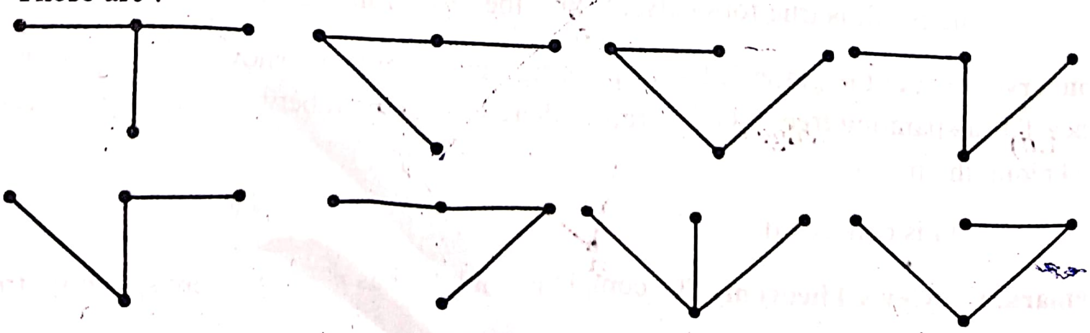
Number of ways for removing 2 edges $= {}^5C_2 = 10$ ways.

But removal of edges should not disconnect graph.

If we remove e_1, e_2 then graph is disconnected. Similarly removal of e_3, e_4 result in disconnected graph.

$\therefore$ there are 8 possible spanning trees of given graph.

These are :

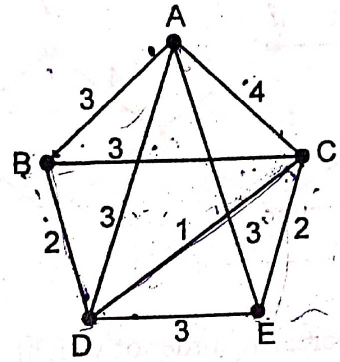


Art-8. Kruskal's Algorithm to find Minimal Spanning Tree

Let G be the given connected graph with n vertices. Then Kruskal Algorithm to find minimal spanning tree involves following steps :

1. Write all the edges of graph in increasing order of their weight.
2. Select the smallest edge of G.
3. For each successive step select another smallest edge of G which makes no cycle with previously selected edges.
4. Go on repeating step 3 until $n-1$ edges have been selected. The sum of weights of these $n-1$ edges will constitute required minimal spanning tree.

Example 19. Find the minimal spanning tree of weighted graph using Kruskal's algorithm.



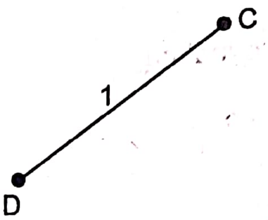
Sol. Number of vertices $(n) = 5$

First we write all edges in increasing order of weight

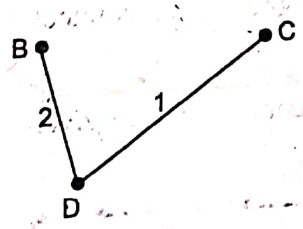
E = {CD, BD, CE, AB, BC, AD, AE, DE, AC}

We start from edge CD and then select edges one by one from E until we select 4 edges $(n-1)$.

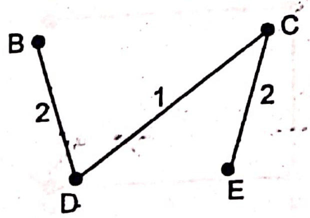
(i)



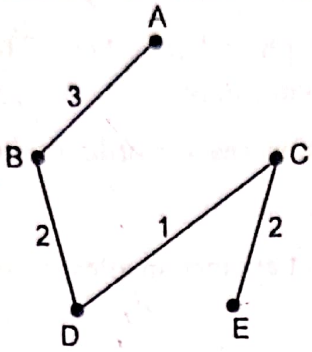
(ii) Select next edge BD



(iii) Select next edge CE

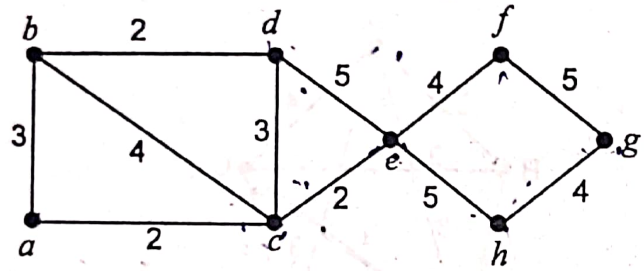


(*iv*) Select next edge AB



Since we have 5 vertices and we have selected 4 edges, so we stop algorithm. Minimal spanning tree is as shown and sum of weights is $1+2+2+3=8$.

Example 20. Find the minimal spanning tree for the following weighted connected graph using Kruskal's Algorithm.



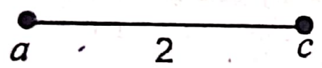
Sol. First we write all edges in Increasing order of weight

$$E = \{ac, bd, ce, ab, cd, bc, ef, gh, ed, eh, fg\}$$

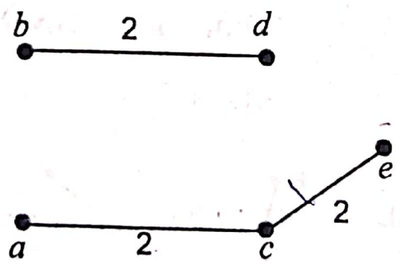
Number of vertices $(n) = 8$

We start from edge ac and then select edges one by one from E until we select 7 edges $(n-1)$

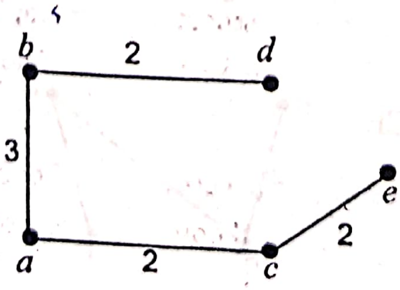
(*i*)



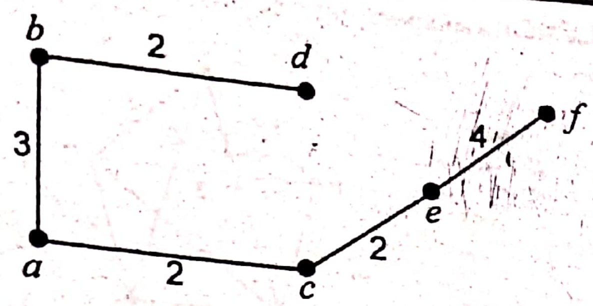
(*ii*) Select next edge bd and then ce.



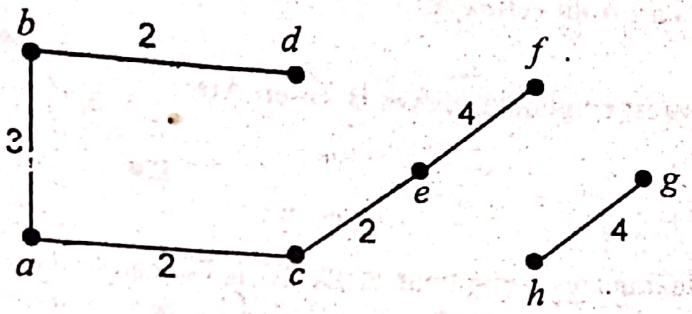
(*iii*) Select next edge ab



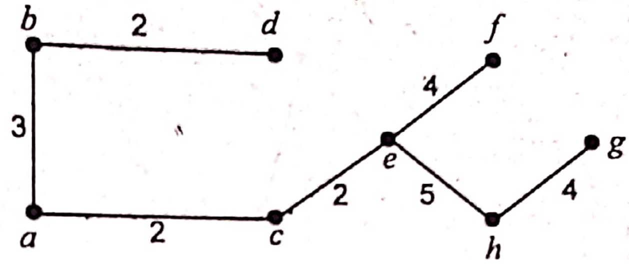
(iv) Select next edge *ef*



(v) Select next edge *gh*



(vi) Select next edge *eh*



As we have selected 7 edges, so we stop algorithm. Minimal spanning tree is as own and sum of weights is

$$2 + 2 + 2 + 3 + 4 + 4 + 5 = 22.$$

rt-9. Prim's Algorithm to Find Minimal Spanning Tree

Let G be the given graph with n vertices. Then Prim's algorithm to find minimal anning tree involves following steps :

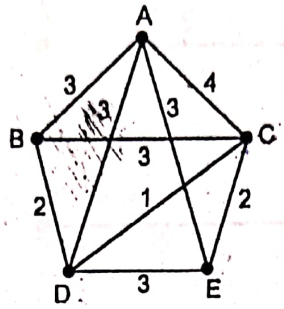
1. Choose any vertex V_1 of G or start from given vertex.

2. Connect V_1 to its nearest neighbour say V_i.

3. Taking (v_1, v_i) as one subgraph, connect this subgraph to its nearest neighbour *i.e.* ertex which is nearest to V_1 or V_i. Let this vertex is V_k. The new vertex must not form a ycle with previous added vertices.

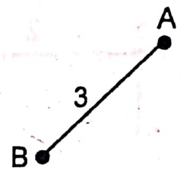
4. Go on repeating step 3 until all n vertices have been connected by $n-1$ edges. he sum of weights of these $n-1$ edges will constitute required minimal spanning tree.

Example 21. Find minimal spanning tree of weighted graph using Prim's algorithm.

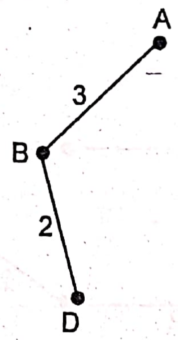


Sol. Let us start from vertex A.

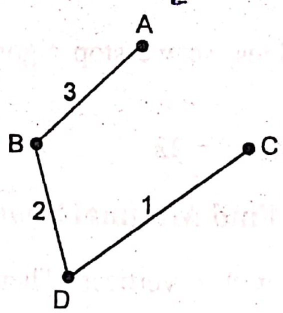
(*i*) Nearest neighbour of A is B. Insert AB.



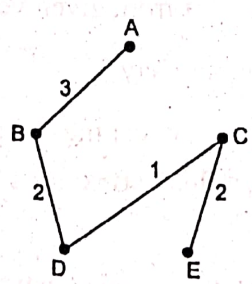
(*ii*) Next nearest neighbour of {A, B} is D. Insert BD.



(*iii*) Next nearest neighbour of {A, B, D} is C. Insert CD.



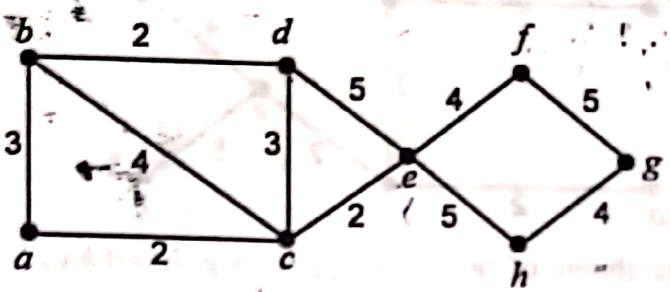
(*iv*) Next nearest neighbour of {A, B, D, C} is E. Insert CE.



As number of vertices are 5 and we have connected 5 vertices using 4 edges so we stop algorithm. Sum of weights is $= 1 + 2 + 2 + 3 = 8$.

Example 22. Find minimal spanning tree for the following weighted connected graph using Prim's algorithm by starting at e.

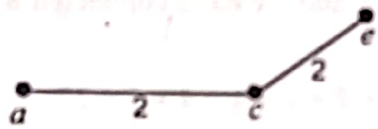
(B.C.A.-II, April, 2007)



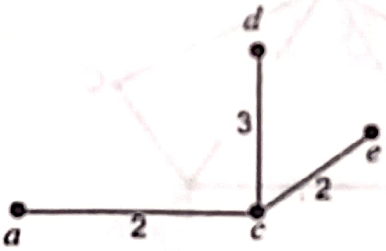
Sol. We start from given vertex e.

(*i*) Nearest neighbour of e is c. Insert $e\,c$.

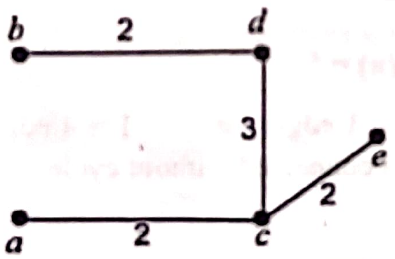
(*ii*) Next nearest neighbour of $\{e, c\}$ is a. Insert $c\,a$.



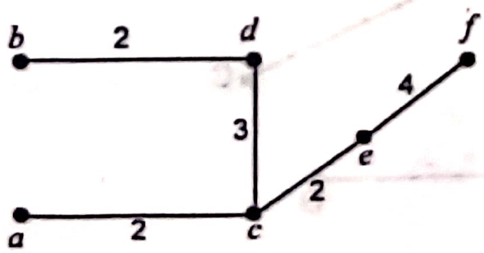
(*iii*) Next nearest neighbour of $\{e, c, a\}$ is d. Insert $c\,d$.



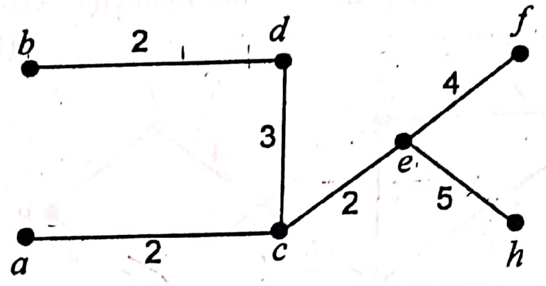
(*iv*) Next nearest neighbour of $\{e, c, a, d\}$ is b. Insert $b\,d$.



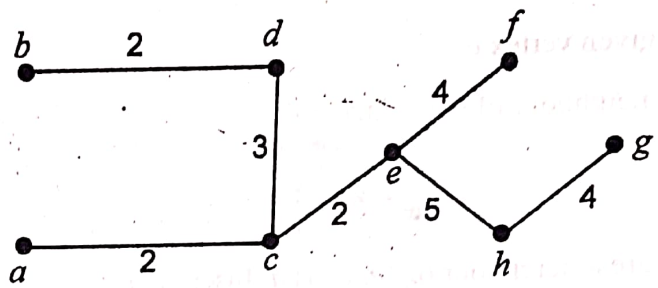
(*v*) Next nearest neighbour of $\{e, c, a, d, b\}$ is f. Insert $e\,f$.



(vi) Next nearest neighbour of $\{e, c, a, d, b, f\}$ is h. Insert $e\,h$.



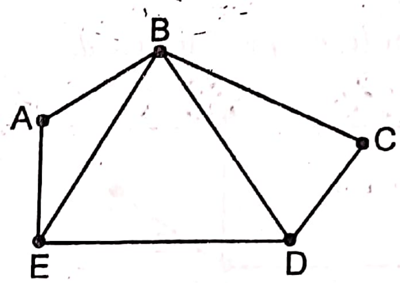
(vii) Next nearest neighbour of $\{e, c, a, d, b, f, h\}$ is g. Insert $h\,g$.



As number of vertices are 8 and we have connected 8 vertices using 7 edges, so we stop algorithm.

Sum of weights is $= 2 + 3 + 2 + 2 + 4 + 4 + 5 = 22$.

Example 23. Generate a spanning tree for the graph.



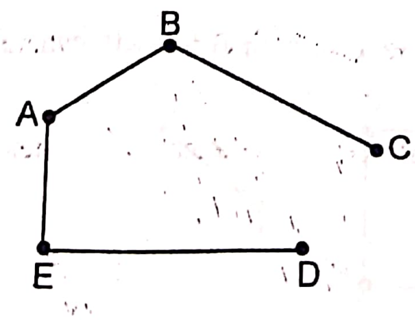
(P.T.U. B.C.A.-I 2005)

Sol. Number of edges in graph = 7

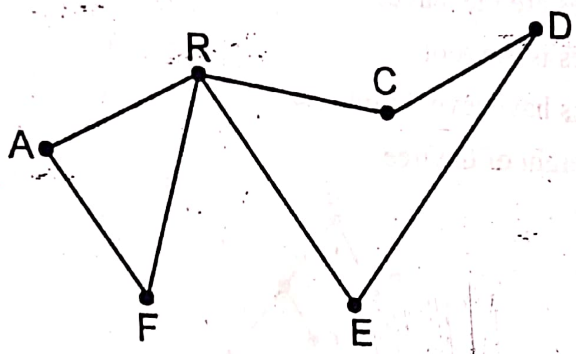
Number of vertices in graph $(n) = 5$

For spanning tree we need $n - 1$ edges *i.e.* $5 - 1 = 4$ edges that must be connected in such a way so that graph must be connected without cycle.

One such spanning tree is :



Example 24. Generate a spanning tree for :



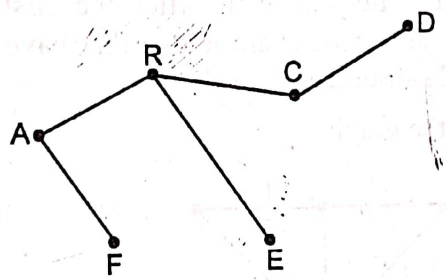
(P.T.U. B.C.A.-I, 2005)

Sol. Number of edges in graph = 7

Number of vertices in graph = 6 (n)

For spanning tree we need $n - 1$ edges *i.e.* $6 - 1 = 5$ edges that must be connected in such a way so that graph must be connected without cycle.

One such spanning tree is :



EXERCISE 8 (*a*)

1. Draw all the trees consisting of

   (*a*) One vertex (*b*) Two vertices

   (*c*) Three vertices (*d*) Four vertices

   (*e*) Five vertices (*f*) Six vertices.

2. If $T_1 = (V_1, E_1)$, $T_2 = (V_2, E_2)$ be two trees where $|E_1| = 17$ and $|V_2| = 2 \cdot |V_1|$. Find $|V_1|$, $|V_2|$ and $|E_2|$.

3. If $F_1 = (V_1, E_1)$ be forest of 7 trees where $|E_1| = 40$ then what is $|V_1|$?

4. If $F_2 = (V_2, E_2)$ be forest with $|V_2| = 62$ and $|E_2| = 51$, how many trees determine F_2?

5. Give an example of an undirected graph $G = (V, E)$, where $|E| = |V| - 1$, but G is not a tree.

6. Prove that in any non-trivial tree there is atleast one vertex of degree 1.

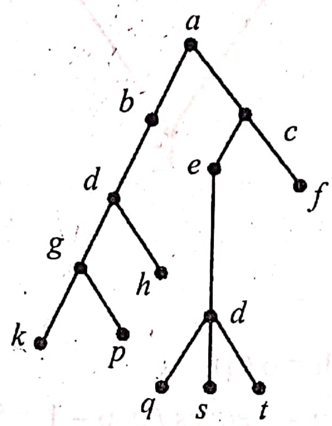
7. Answer the following questions for the tree shown below :

(a) Which vertices are the leaves

(b) Which vertices is the root

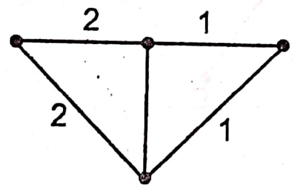
(c) Which vertices have level number 4

(d) What is the height of the tree

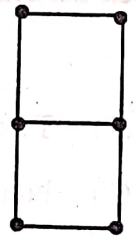


8. On the first Sunday of 1993 Ram and Shyam start a chain letter, by sending 3 letters. Each person receiving the letter is to send 3 copies to 3 new people on the Sunday following the letter's arrival. After the first five Sundays have passed, what is the total number of chain letters that have been mailed ? How many were mailed on the last Sunday.

9. Find all spanning trees of the graph.

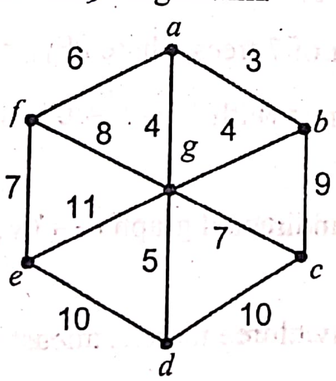


10. Find the number of spanning trees of the figure given below.



(Pbi. U. 2009)

11. Find a minimal spanning tree for the connected weighed graph given below using both Kruskal's and Prim's algorithm.

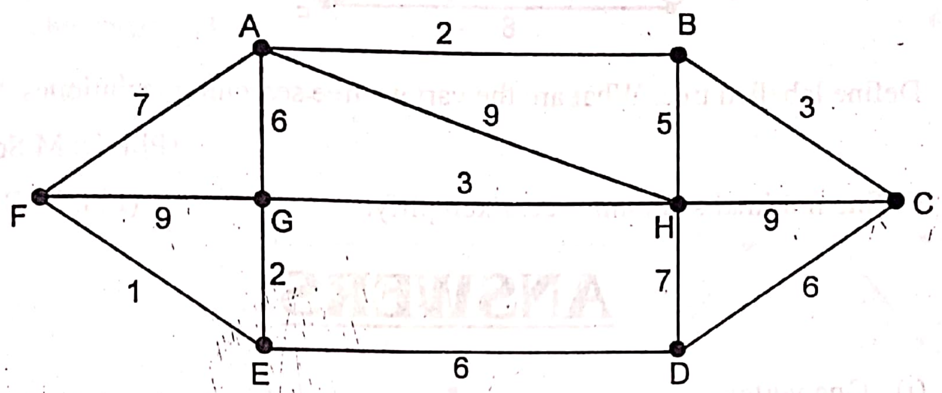


12. Draw all rooted tree with 5 nodes.

13. Draw all binary tree with 4 leaves.

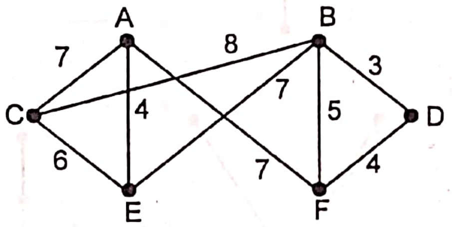
14. Draw all binary tree with 6 leaves.

15. Use Kruskal algorithm to find spanning tree of minimal weight by showing each step.



16. Find minimal spanning tree of above problem using Prim's algorithm by starting from D.

17. Find minimal spanning tree of weighted graph shown below



18. Show that maximum number of vertices in a binary tree of height n is $2^{n+1} - 1$.

19. Prove that largest number of leaves in an n-tree of height k is n^k.

20. Construct expression tree of following :

(*a*) $(x \div y) \div ((x \times 3) - (z \div 4))$

(*b*) $((2 \times x) + (3 - (4 \times x))) + (x - (3 \times 11))$

(*c*) $(((2 \times 7) + x) \div y) \div (3 - 11)$.

(*d*) $(3 - (2 - (11 - (9 - 4)))) \div (2 + (3 + (4 + 7)))$

(Pbi.U. M.Sc. I.T. 2010, 2011)

21. Evaluate the expressions given in polish notation

(*a*) $\times \; - \; + \; 3 \; 4 \; - 7 \; 2 \div \; 12 \; \times 3 \; - \; 6 \; 4$

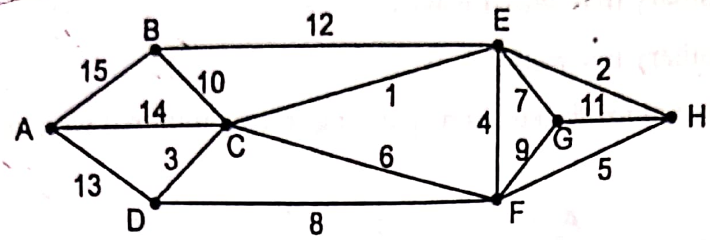
(*b*) $\div \; - \times 3 \; x \times \; 4 \; y \; + 15 \; \times 2 - 6 \; y$ where x is 2 and y is 3.

22. Evaluate the expression given in reverse polish notation

$7 \; x \; \times y - 8 \; x \times w \; + \; \times$ where x is 7, y is 2 and w is 1.

23. What do you mean by Tree Searching? Discuss various algorithms for searching the trees? (Pbi. U. BCA 2009)

24. Define a Minimum spanning. Tree of a graph and find the same for the graph.



25. Define labelled tree. What are the various tree searching techniques ?

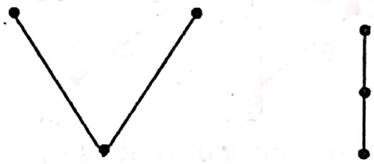
(Pbi. U. M.Sc. I.T. 2011)

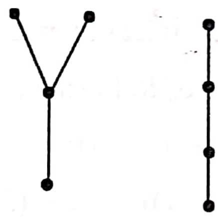
26. Define minimal spanning tree. Exemplify. (Pbi. U. B.C.A. 2012)

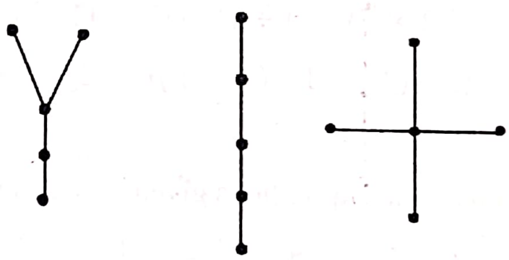
ANSWERS

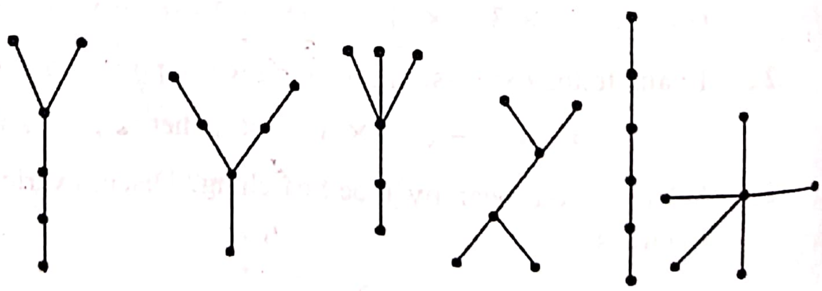
1. (i) One vertex

(ii) Two vertices 

(iii) Three vertices 

(iv) Four vertices 

(v) Five vertices 

(vi) Six vertices 

2. 18, 36, 35 3. 47 4. 11

7. (*a*) The vertices f, h, k, p, q, s, t are the leaves

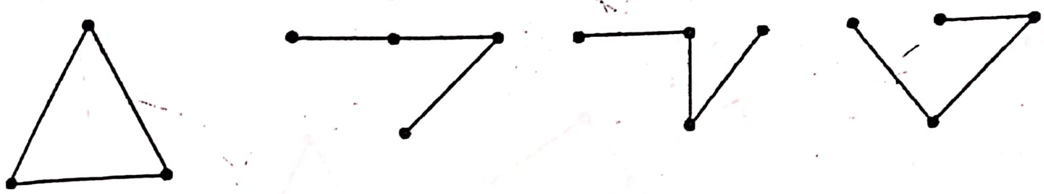
(*b*) The vertex a is the root

(*c*) The vertices k, p, q, s, t are at level 4

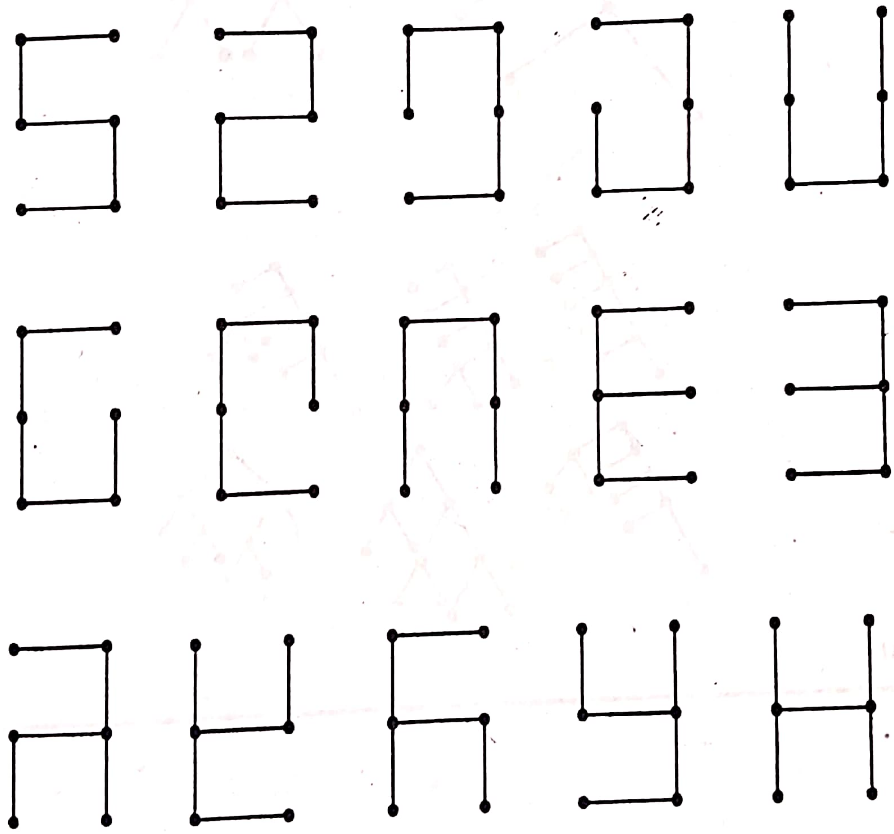
(*d*) The height of the tree is 4

8. 363, 243

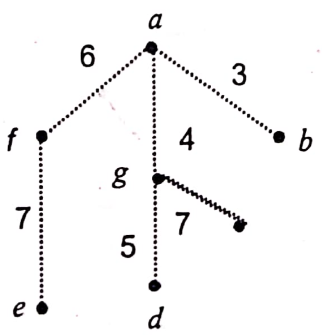
9.



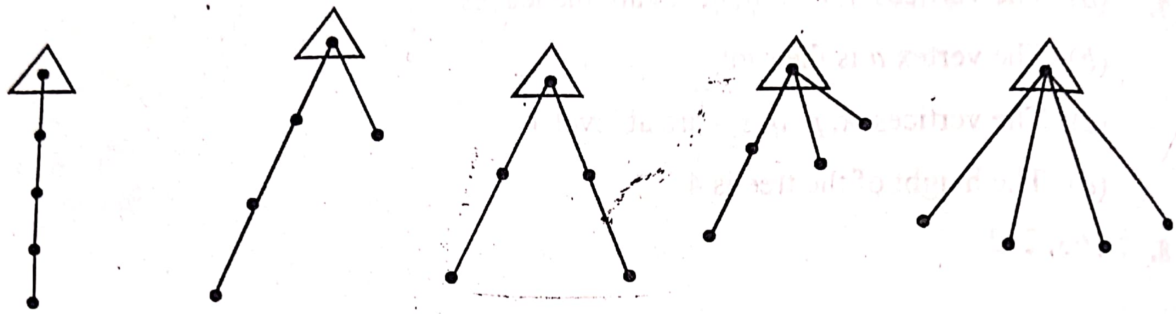
10.



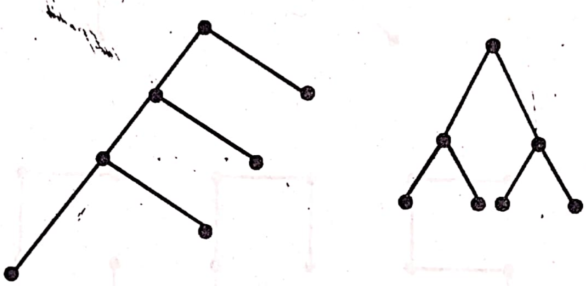
11.



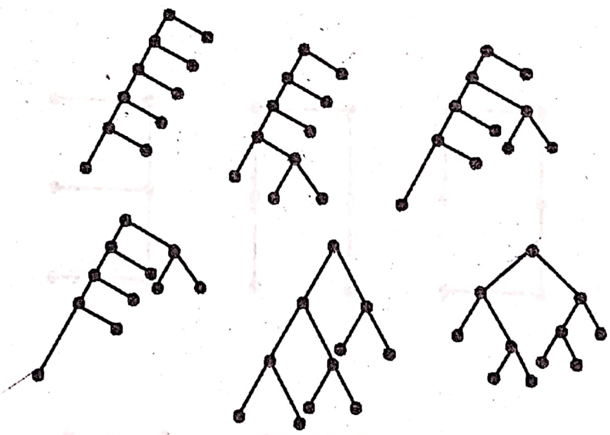
12.



13.



14.



24. 40