12 TREES

12.1. INTRODUCTION

In this chapter, we will discuss a special class of graphs, called trees. The concept of trees is frequently used in both mathematics and sciences. To understand the concept of trees, it is essential to know the various common types of trees. Their basic properties and applications.

12.2. TREE

A graph which has no cycle is called an acyclic graph. A tree is an acyclic graph or graph having no cycles.

A tree or general tree is defined as a non-empty finite set of elements called vertices or nodes having the property that each node can have minimum degree 1 and maximum degree n. It can be partitioned into $n + 1$ disjoint subsets such that the first subset contains the root of the tree and the remaining n subsets contains the elements of the n subtree. (Fig. 12.1)

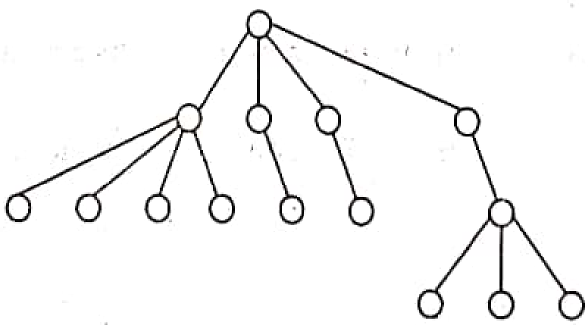


Fig. 12.1. General Tree.

12.3. DIRECTED TREES

A directed tree is an acyclic directed graph. It has one node with indegree 1, while all other nodes have indegree 1 as shown in Figs. 12.2 and 12.3.

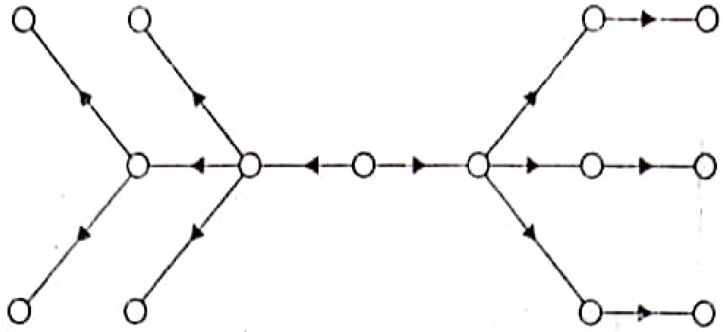


Fig. 12.2

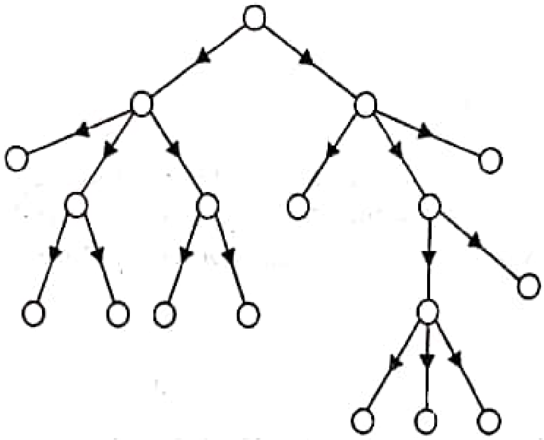


Fig. 12.3

The node which has outdegree 0 is called an external node or a terminal node or a leaf. The nodes which has outdegree greater than or equal to one are called internal nodes or branch nodes.

12.4. ORDERED TREES

If in a tree at each level, an ordering is defined, then such a tree is called an ordered tree.

e.g., the trees shown in Figs. 12.4 and 12.5 represent the same tree but have different orders.

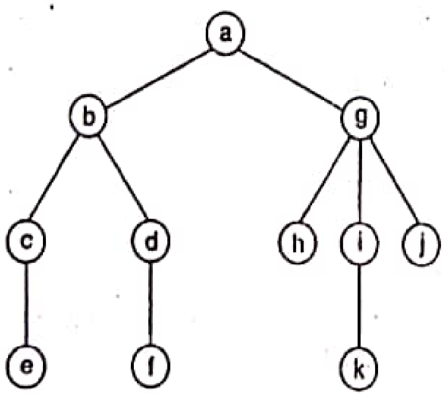


Fig. 12.4

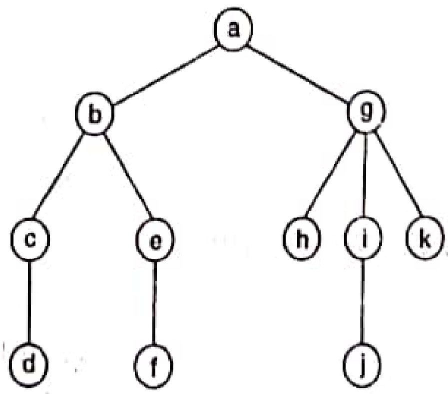


Fig. 12.5

12.5. ROOTED TREES

If a directed tree has exactly one node or vertex called root whose incoming degree is 0 and all other vertices have incoming degree one, then the tree is called rooted tree.

* A tree with no nodes is a rooted tree (the empty tree).
* A single node with no children is a rooted tree.

Example. *Suppose 8 people enter a Badminton tournament use a rooted tree model of the tournament to determine how many games must be played to determine a champion if a player is eliminated after one loss.* (P.T.U. B.Tech. May 2009)

Sol. As there are 8 people in the badminton tournament, there will be four games to be played in the first round two games to be played in the second round, one game to be played in the final round

Hence, total number of games in the tournament is 7. (See Fig. 12.6).

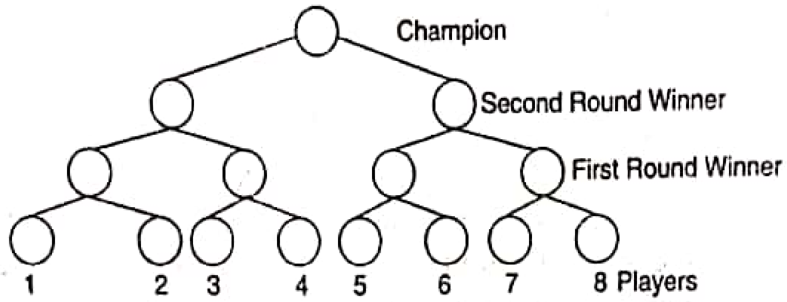


Fig. 12.6

12.6. PATH LENGTH OF A VERTEX

The path length of a vertex in a rooted tree is defined to be the number of edges in the path from the root to the vertex.

For example, we find the path lengths of the nodes b, f, l, q in Fig. 12.7a.

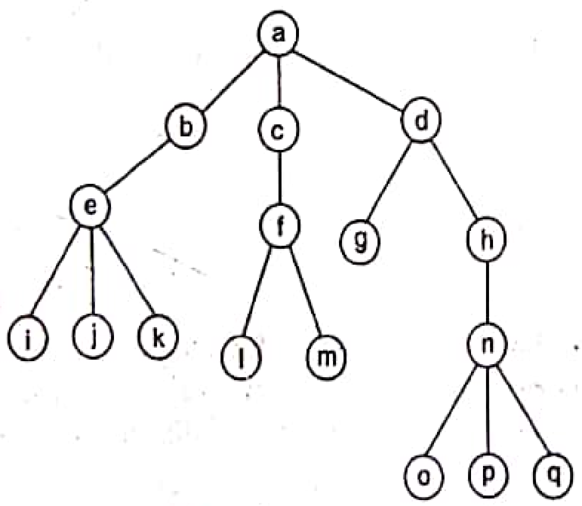


Fig. 12.7*a*

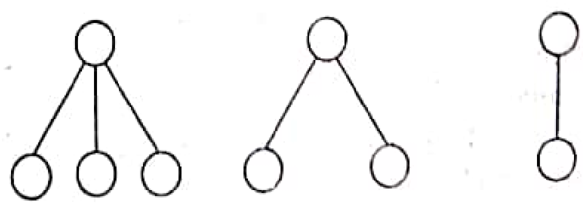


Fig. 12.7*b*

The path length of node b is one.
The path length of node f is two.
The path length of node l is three.
The path length of node q is four.

Theorem I. *Prove that there is one and only one path between every pair of vertices in a tree T.*

Proof. We know that T is a connected graph, in which there must exist at least one path between every pair of vertices. Now assume that there exists two different paths from some node a to some node b of T. The union of these two paths will contain a cycle and therefore T cannot be a tree. Hence, there is only one path between every pair of vertices in a tree.

12.7. FOREST

If the root and the corresponding edges connecting the nodes are deleted from a tree, we obtain a set of disjoint trees. This set of disjoint trees is called a forest. (Fig. 12.7*b*)

12.8. BINARY TREE

If the outdegree of every node is less than or equal to 2, in a directed tree then the tree is called a binary tree. A tree consisting of no nodes (empty tree) is also a binary tree.

12.9. BASIC TERMINOLOGY

(*a*) **Root.** A binary tree has a unique node called the root of the tree.

(*b*) **Left Child.** The node to the left of the root is called its left child.

(*c*) **Right Child.** The node to the right of the root is called its right child.

(*d*) **Parent.** A node having left child or right child or both is called parent of the nodes.

(*e*) **Siblings.** Two nodes having the same parent are called siblings.

(*f*) **Leaf.** A node with no children is called a leaf. The number of leaves in a binary tree can vary from one (minimum) to half the number of vertices (maximum) in a tree.

(*g*) **Ancestor.** If a node is the parent of another node, then it is called ancestor of that node. The root is an ancestor of every other node in the tree.

(*h*) **Descendent.** A node is called descendent of another node if it is the child of the node or child of some other descendent of that node. All the nodes in the tree are descendents of the root.

(*i*) **Left Subtree.** The subtree whose root is the left child of some node is called the left subtree of that node.

(*j*) **Right Subtree.** The subtree whose root is the right child of some node is called the right subtree of that node.

(*k*) **Level of a Node.** The level of a node is its distance from the root. The level of root is defined as zero. The level of all other nodes is one more than its parent node. The maximum number of nodes at any level N is 2^N.

(*l*) **Depth or Height of a Tree.** The depth or height of a tree is defined as the maximum number of nodes in a branch of tree. This is one more than the maximum level of the tree *i.e.*, the depth of root is one. The maximum number of nodes in a binary tree of depth d is $2^d - 1$, where $d \geq 1$.

(*m*) **External Nodes.** The nodes which has no children are called external nodes or terminal nodes.

(*n*) **Internal Nodes.** The nodes which has one or more than one children are called internal nodes or non-terminal nodes.

Theorem II. *Let G be a graph with more than one vertex. Then the following are equivalent :*

(*i*) *G is a tree.*

(*ii*) *Each pair of vertices is connected by exactly one simple path.*

(*iii*) *G is connected, but if any edge is deleted then the resulting graph is not connected.*

(iv) G is cycle tree, but if any edge is added to the graph then the resulting graph has exactly one cycle.

Proof. To prove this theorem, we prove that (*i*) ⇒ (*ii*), (*ii*) ⇒ (*iii*), (*iii*) ⇒ (*iv*) and finally (*iv*) ⇒ (*i*). The complete proof is as follows :

(*i*) ⇒ (*ii*) Let us assume two vertices u and v in G. Since G is a tree, so G is connected and there is at least one path between u and v. More over, there can be only one path between u and v. otherwise G will contain a cycle.

(*ii*) ⇒ (*iii*) Let us delete an edge $e = (u, v)$ from G. It means e is a path from u to v. Suppose the graph result from $G - e$ has a path p from u to v. Then P and e are two distinct paths from u to v, which is a contradiction of our assumption. Thus, there does not exist a path between u and v in $G - e$, so $G - e$ is disconnected.

(*iii*) ⇒ (*iv*) Let us suppose that G contains a cycle c which contains an edge $e = \{u, v\}$. By hypothesis, G is connected but $G' = G - e$ is disconnected with u and v belonging to different components of G′. This contradicts the fact that u and v are connected by the path $P = C - e$, which lies in G′. Hence G is cycle free.

Now, Let us take two vertices x and y of G and let H be the graph obtained by adjoining the edge $e = \{x, y\}$ to G. Since G is connected, there is a path P from x to y in G ; hence $C = Pe$ forms a cycle in H. Now suppose H contains another cycle C_1. Since G is cycle free, C_1 must contain the edge e, say $C_1 = P_1 e$.

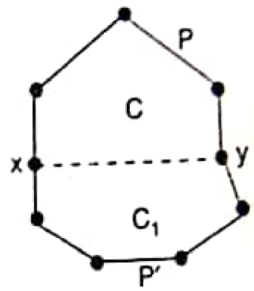


Fig. I

Then P and P_1 are two paths in G from x to y as shown in Fig. I. Thus, G contains a cycle, which contradicts the fact that G is cycle free. Hence H contains only one cycle.

(*iv*) ⇒ (*i*) By adding any edge $C = (x, y)$ to G produces a cycle, the vertices x and y must be connected already in G. Thus, G is connected and is cycle is free *i.e.*, G is a tree.

Theorem III. *Let G be a finite graph with $n > 1$ vertices. Then the following are equivalent :*

(i) G is a tree.

(ii) G is cycle free and has $n - 1$ edges.

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

To prove this.

Proof. We use induction on the number of vertices n of G.

Let us assume $n = 1$ *i.e.*, G has only one vertex. Then G has O edges and so G is connected and cycle free. Thus the theorem holds for $n = 1$.

Now, assume that $n > 1$ *i.e.*, G has more than one vertex. Assume that (*i*), (*ii*) and (*iii*) are equivalent for all graphs with less than n vertices.

We have to show that they are equivalent for G.

(*i*) ⇒ (*ii*) Suppose G is a tree. Then G is cycle free, so we have to show only that G has $n - 1$ edges. We know that G has a vertex of degree 1. Deleting this vertex and its edge, we obtain a tree T which has $n - 1$ vertices. Thus the theorem holds for T, so T has $n - 2$ edges. Hence G has $n - 1$ edges.

(*ii*) ⇒ (*iii*) Suppose G is cycle free and has $n - 1$ edges. We have to show only that G is connected. Suppose G is disconnected and has k components, $T_1, T_2, ..., T_k$, which are trees since each is connected and cycle free, *i.e.*, Ti has ni vertices and $ni < n$. Hence the theorem holds for Ti, so Ti has $ni - 1$ edges. Thus,

$$n = n_1 + n_2 + \dots + n_k$$

and
$$n - 1 = (n_1 - 1) + (n_2 - 1) + \dots + (n_k - 1)$$
$$= n_1 + n_2 + \dots + n_k - k = n - k$$

Hence $k = 1$. But it contradicts our assumption that G is disconnected and has $k > 1$ components. Hence G is connected.

(*iii*) $\Rightarrow$ (*i*) Suppose G is connected and has $n - 1$ edges. We have to show only that G is cycle free. Suppose G has a cycle containing an edge e. Deleting e, we obtain the graph H = G – e, which is also connected.

But H has n vertices and $n - 2$ edges and therefore must be disconnected. Thus G is cycle free and hence it is a tree.

ILLUSTRATIVE EXAMPLES

Example 1. *For the tree as shown in Fig. 12.8.*

(*i*) *Which node is the root ?*

(*ii*) *Which nodes are leaves ?*

(*iii*) *Name the parent node of each node.*

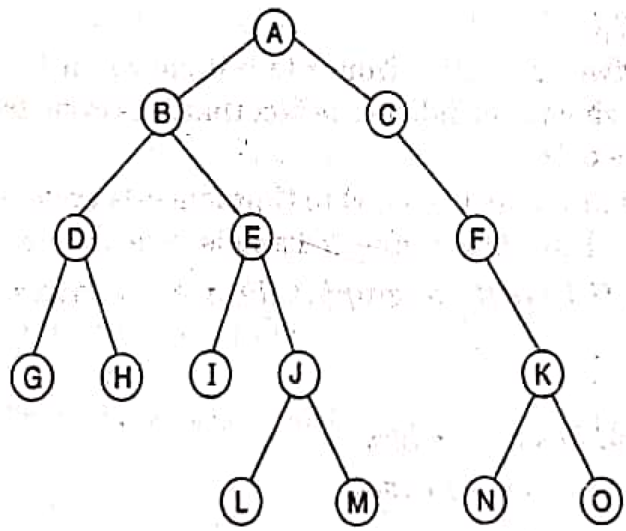


Fig. 12.8

Sol. (*i*) The node A is the root node.

(*ii*) The nodes G, H, I, L, M, N, O are leaves.

(*iii*)

Nodes	Parent
B, C	A
D, E	B
F	C
G, H	D
I, J	E
K	F
L, M	J
N, O	K

Example 2. *For the tree as shown in Fig. 12.9.*

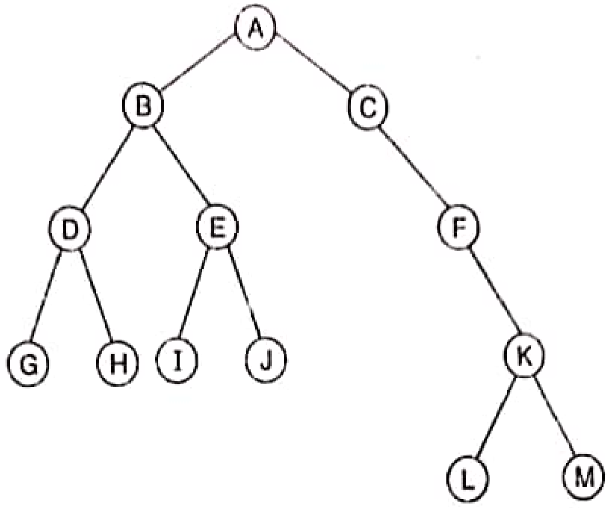


Fig. 12.9

(i) List the children of each node. *(ii) List the siblings.*
(iii) Find the depth of each node. *(iv) Find the level of each node.*

Sol. (*i*) The children of each node is as follows :

Node	Children
A	B, C
B	D, E
C	F
D	G, H
E	I, J
F	K
K	L, M

(*ii*) The siblings are as follows :

Siblings
B and C
D and E
G and H
I and J
L and M are all siblings.

(*iii*)

Node	Depth or Height
A	1
B, C	2
D, E, F	3
G, H, I, J, K	4
L, M	5

(*iv*)

Node	Level
A	0
B, C	1
D, E, F	2
G, H, I, J, K	3
L, M	4

Imp

Example 3. *Show that if in a graph G there exists one and only one path between every pair of vertices, then G is a tree.*

Sol. The graph G is connected since there is a path between every pair of vertices. A cycle in a graph exists if there is at least one pair of vertices (v_1, v_2) such that there exist two distinct paths from v_1 to v_2. But the graph G has one and only one path between every pair of vertices. Thus, G contains no cycle. Hence, G is a tree.

Example 4. *Draw two different binary trees with five nodes having only one leaf.*

Sol. The two trees out of many possible trees with five nodes having only one leaf is shown in Fig. 12.10.

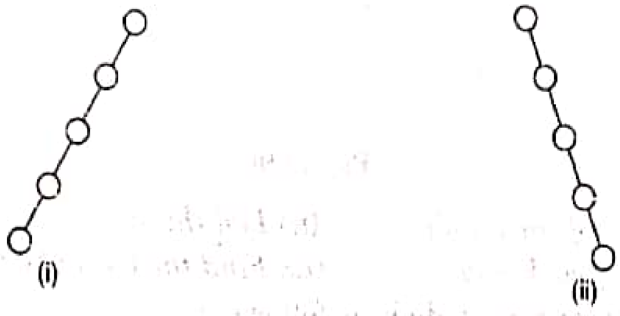


Fig. 12.10

Example 5. (*a*) *Draw two different binary trees with five nodes having maximum number of leaves.*

(*b*) *Let T be a tree with n vertices. Determine the number of leaf nodes in a tree.*

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

Sol. (*a*) There are many possible trees, out of which two different binary trees are shown in Fig. 12.11.

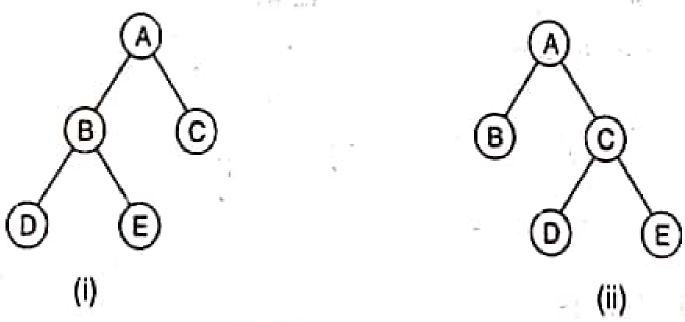


Fig. 12.11

(*b*) Given T is a binary tree with n vertices. Therefore, the tree with n vertices has $(n-1)$ edges. Also, if L denotes the number of leaves and I be the number of internal nodes, then

$$n = L + 1$$

But $$I = \frac{n-1}{2}$$

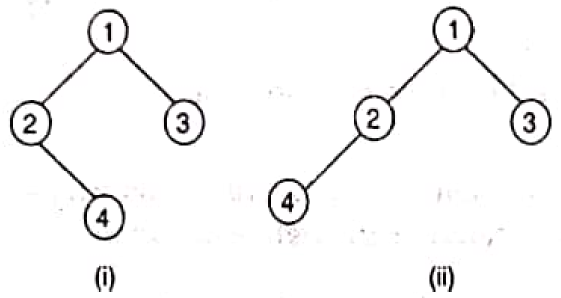
Using (2) in (1), we have

$$n = L + \frac{n-1}{2} \Rightarrow L = n - \frac{n-1}{2} = \frac{2n-n+1}{2} = \frac{n+1}{2}$$

Example 6. (*a*) *How will you differentiate between a general tree and a binary tree ?*

(*b*) *Define a rooted tree with an example and show how it may be viewed as directed graph.*

Sol. (*a*)

General Tree	*Binary Tree*
1. There is no such tree having zero nodes or an empty general tree.	1. There may be an empty binary tree.
2. If some node has a child, then there is no such distinction.	2. If some nodes has a child, then it is distinguished as a left child or a right child.
3. The trees shown in Fig. 12.12 are same, when we consider them as general trees.	3. The trees shown in Fig. 12.12 are distinct, when we consider them as binary trees, because in (*i*), 4 is right child of 2 while in (*ii*), 4 is left child of 2.
 Fig. 12.12	

(*b*) **Rooted tree:** We first define the term '*directed tree*'. A directed graph is said to be a directed tree if it becomes a tree when the directions of the edges are ignored. For example, the Fig. 12.13 is a *directed tree*.

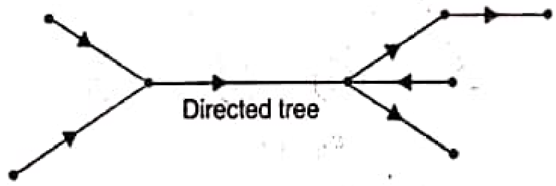


Fig. 12.13

A directed tree is called a *rooted tree* if there is exactly one vertex whose incoming degree is 0 and incoming degree of all other vertices are 1. The vertex with incoming degree 0 is called the *root* of the *rooted tree*. The Fig. 12.14 is an example of a rooted tree.

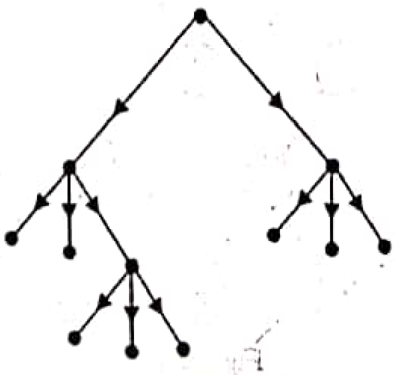


Fig. 12.14

In a rooted tree, a vertex whose outgoing degree is 0 is called a *leaf* or a *terminal code* and a vertex whose *outgoing degree is non zero, is called a branch node or an internal node.*

Rooted tree may be viewed as directed graph. We know that a tree is a graph which is connected and without any cycles. A rooted tree T is a tree with a designated vertex *r*, called the root of the tree. Since there is a unique simple path from the root *r* to any other vertex *v* in T, this determines a direction to the edges of T. Thus T may be viewed as a directed graph.

12.10. BINARY EXPRESSION TREES

Algebraic expression can be conveniently expressed by its expression tree. An expression having binary operators can be decomposed into

< left operand or expression > (operator) < right operand or expression >

depending upon precedence of evaluation.

The expression tree is a binary tree whose root contains the operator and whose left subtree contains the left expression and right subtree contains the right expression.

Example 7. *Construct the binary expression tree for the expression (a + b) * (d/c).*

Sol. The binary expression tree for the expression $(a + b) * (d/c)$ is shown in Fig. 12.15.

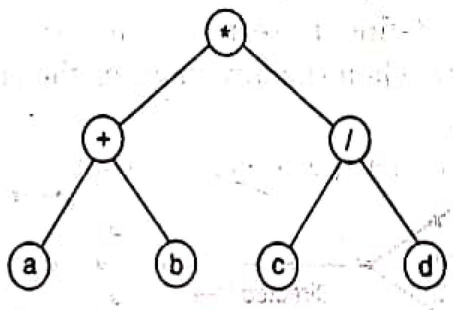


Fig. 12.15

Example 8. *Determine the value of expression tree shown in Fig. 12.16.*

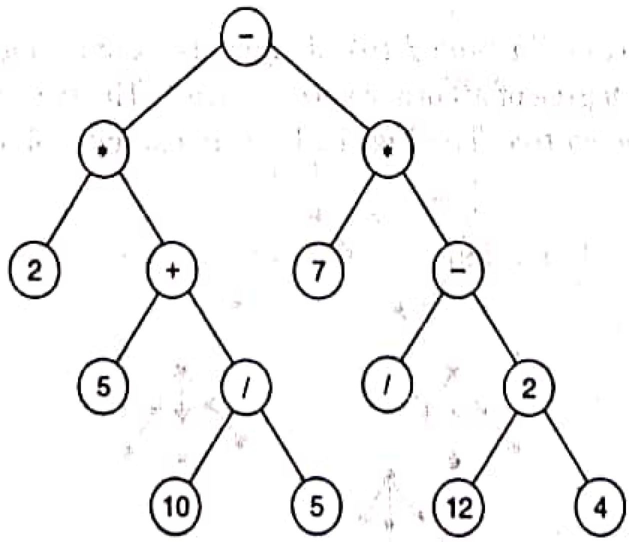


Fig. 12.16

Sol. The value of expression tree is 2.

Example 9. *Determine the value of expression tree shown in Fig. 12.17.*

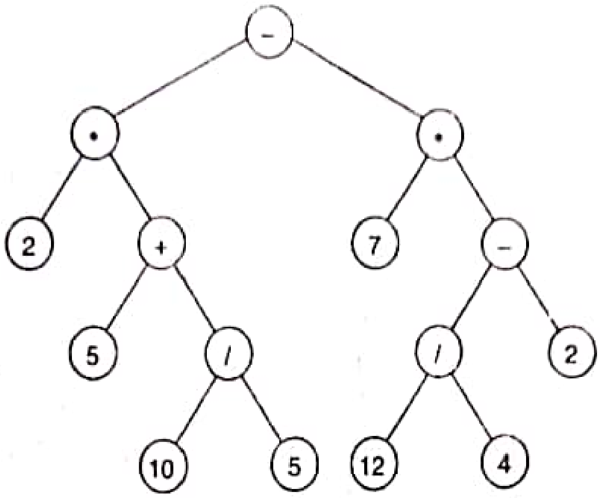


Fig. 12.17

Sol. The value of expression tree is 7.

12.11. COMPLETE BINARY TREE

Complete binary tree is a binary tree if all its levels, except possibly the last, have the maximum number of possible nodes as for left as possible. The depth of complete binary tree having n nodes is $\log_2 n + 1$.

For example : The tree shown in Fig. 12.18 is a complete binary tree.

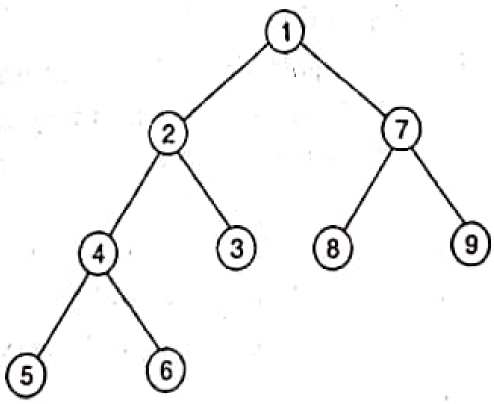


Fig. 12.18

12.12. FULL BINARY TREE

Full binary tree is a binary tree in which all the leaves are on the same level and every non-leaf node has two children.

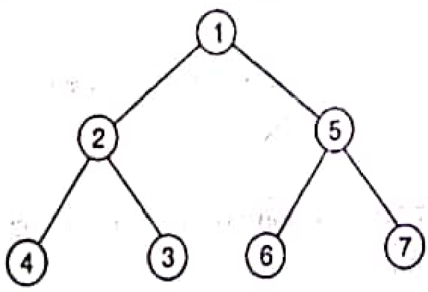


Fig. 12.19

For example : The tree shown in Fig. 12.19 is a full binary tree.

Theorem I. *Prove that the maximum number of nodes on level n of a binary tree is 2^n, where $n \geq 0$.*

Proof. This can be proved by induction.

Basis of Induction. The only node at level $n = 0$ is the root node. Thus, the maximum number of nodes on level $n = 0$ is $2^0 = 1$.

Induction Hypothesis. Now assume that it is true for level j, where $n \geq j \geq 0$. Therefore, the maximum no. of nodes on level j is 2^j.

Induction Step. By induction hypothesis, the maximum number of nodes on level $j - 1$ is 2^{j-1}. Since, we know that each node in binary tree has maximum degree 2. Therefore, the maximum number of nodes on level j is twice the maximum number of level $j - 1$.

Hence, at level j, the maximum number of nodes is

$$= 2 . 2^{j-1}$$

$$= 2^j \text{. Hence proved.}$$

Theorem II. *Prove that the maximum number of nodes in a binary tree of depth d is $2^d - 1$, where $d \geq 1$.*

Proof. This can be proved by induction.

Basis of Induction. The only node at depth $d = 1$ is the root node. Thus, the maximum number of nodes on depth $d = 1$ is $2^1 - 1 = 1$.

Induction Hypothesis. Now assume that it is true for depth K, $d > k \geq 1$. Therefore, the maximum number of nodes on depth K is $2^k - 1$.

Induction Step. By induction hypothesis, the maximum number of nodes on depth $K - 1$ is $2^{k-1} - 1$. Since, we know that each node in a binary tree has maximum degree 2, therefore, the maximum number of nodes on depth $d = K$ is twice the maximum number of nodes on depth $K - 1$.

So, at depth $d = K$, the maximum number of nodes is $= (2 . 2^{K-1}) - 1$

$$= 2^{K-1+1} - 1 = 2^K - 1 \text{. Hence proved.}$$

Theorem III. *Prove that in a binary tree, if n_E is the number of external nodes or leaves and n_I is the no. of internal nodes, then $n_E = n_I + 1$.*

Proof. Let n be the total number of nodes in the tree. Then, we may have three types of nodes in the tree.

n_E = the number of nodes having zero degree.

n_1 = the number of nodes having two degree.

n_0 = the number of nodes having one degree.

So, we have $\quad n = n_E + n_1 + n_0 \quad$ ...(1)

Let us assume that the number of edges of the tree is E. So, with these E edges we can connect E + 1 nodes.

Hence $\quad n = E + 1 \quad$ ...(2)

Since, all edges are either from a node of degree one or from a node of degree two, therefore, $\quad E = n_0 + 2n_1 \quad$ ...(3)