Introduction:
Tree is a collection of nodes and among those nodes,one node is
taken as root node.Rest of the nodes are taken as disjoint subsets.
Each subsets is a tree again or a subtree again.

Root -> The first node of the tree.

Parent -> A node is a parent to it's very next descendants.

children -> descendants which are connected by a single edge with a node.

Siblings -> Children of same parents.

Descendants -> they are the sets of nodes which can be reached from
a particular node downwards.

Ancestors -> Between the path of two particular nodes,all other nodes
along the path are called ancestors.

Degree of a node -> Number of direct children of a particular node.

Leaf Nodes -> Nodes with degree 0 are leaf nodes./External nodes./Terminal nodes

Non-Leaf Nodes -> Nodes with degrees > 0 are non-leaf nodes/internal nodes/non-terminal nodes.

Levels -> starts from 1 and Horizontally measured .

Height -> Root is of height 0,So height of a tree starts from 0 onwards till the last reachable node.

Forest -> A collection of tree is a forest.

Binary tree -> Every node can have max true children min 0 children.

Shapes of binary tree -> For n nodes , $2nC_n/n+1$ binary trees can be formed.

Binary trees of max height -> For n nodes, 2^{n-1} binary trees of max height are possible.

Binary trees of max height(labeled nodes) -> For n label nodes,$(2nC_n/n+1)*n!$ binary trees of max height are possible.

Minimum Nodes of a binary tree -> n=h+1 (h=height)

Maximum Nodes of a binary tree -> $n= 2^{h+1} - 1$; (h=height)

Minimum Height of a binary tree -> $\log_2 (n+1)-1$;
Maximum Height of a binary tree -> h=n-1; (n=node)

In Binary tree ,the number of deg(2) = the number of deg(0) + 1

Strict Binary Tree -> The binary tree which can have either 0 child or 2 children.

For Strict Binary Tree ->
Minimum Nodes -> (2*h)+1 (h=height)
Maximum Nodes -> 2^{h+1} - 1
Minimum Height -> $\log_2(n+1)-1$ (n=nodes)
Maximum Height -> (n-1)/2 (n=nodes)
External Nodes(Leaf Nodes)=Internal Nodes(Non-leaf nodes)+1

M-**ary Trees** -> the degree of the tree is M,so in M-ary trees,every nodes of the tree can have from 0 to at most M children.Not more than that.

Strict M-ary Trees -> Every nodes of the tree can have either 0 children or M children.

For M-ary Trees->
Minimum Nodes -> Mh + 1 (M=degree,h=height)
Maximum Nodes -> $(M^{(h+1)}-1)/M-1)$ (M=degree,h=height)
Minimum Height -> h= (n-1)/M
(M=degree,h=height,n=nodes)
Maximum Height -> $\log_M[n(M-1)+1]-1$
(M=degree,h=height,n=nodes)
External Nodes(Leaf Nodes)
=(M-1)*(Internal Nodes(Non-leaf nodes))+1

Full Binary Tree -> If a tree of height h,has max number of nodes,that tree will be called full binary tree.

Complete Binary Tree -> If a binary tree is formed from left to right ,say root then left child , then right child ..following this or it can be said like this: If a binary tree nodes are defining by an array elements ,if from first to last of the array ,there is no

blank space between the elements ,then that will be a complete binary tree.Briefly it can be said that,from root ,then left child,then right child,in every level we have to count from left to right.
During counting there must not be any blank space.That tree will be called complete binary tree then.
Book definition: A complete binary tree of height h is a full binary tree till h-1 level.then the last level is filled with nodes from left to right without any gap.

A full binary tree is always a complete binary tree but a complete binary tree can’t be a full binary tree always.

Tree Traversal

Pre-Order -> Visit(node),Preorder(left subtree), Preorder(right subtree).

In-Order ->Inorder(left subtree),Visit(node), Inorder(right subtree).

Post-Order ->Postorder(left subtree),postorder(right subtree), Visit(node).

Level-Order -> Level by level traversal.