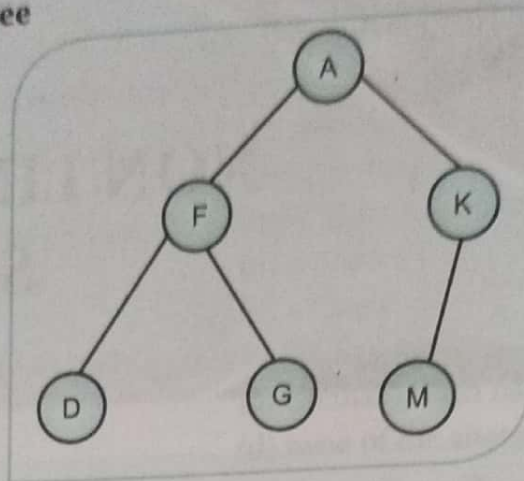
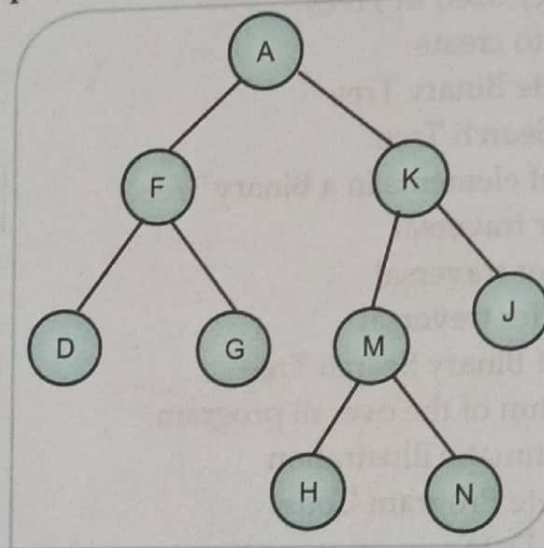


A **Binary Tree** is a special case of a Tree, in which each element contains the path to connect to (at most) two elements.

Diagram of a binary tree



Shown below is a **complete binary tree**. [Every node has out-degree of either 2 or 0.]



### Terminology used in Trees

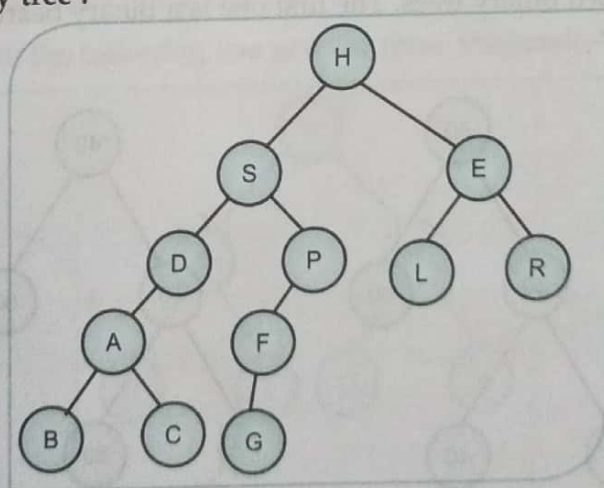
- **Node** : It is a structured data having two parts, "data" and "link to another node".
- **Root Node** : It is the first ( or topmost) node of a tree. It does not have any parent node.
- **External Node/ Leaf Node/Terminal Node** : It is a node that does not have any child node.
- **Internal Node** : It is a node that has child node or nodes.
- **In-degree of a node** : The number of edges that are going in a node is called the in-degree of that node. In-degree of Root is 0. A tree cannot have any node with In-degree > 1.
- **Out-degree of a node** : The number of edges that are coming out of a node is called the out-degree of that node. Out-degree of leaf nodes is 0. A binary tree cannot have any node with Out-degree > 2.
- **Parent and Child Node** – For two nodes, say F and G, if node G comes from node F then,
  - ◆ node F is called the Parent of node G
  - ◆ node G is called the Child of node F



- **Edge** : The connecting line between two nodes is called an Edge. An edge is denoted by using the two respective nodes. E.g., AF, FG, KM etc.
- **Path** : It is the sequence of continuous edges between two nodes. E.g., the path of node A to node G is AFG.
- **Sibling Node** : Nodes that appear from the same parent are called sibling nodes.
- **Sub-tree** : It is a part of a tree in which a node behaves as the root having its own set of descendents (or child nodes). Every node in a tree is capable of becoming a root.
- **Depth of a Node** : It is the number of edges from Root to that Node. Depth of Root is 0.
- **Level of a Node** : It is same as the Depth of a Node. The Level of Root is 0.
- **Height of a Node** : It is the number of edges from the Node to its deepest leaf of its Sub-tree. Height of each leaf is 0. Height of Root is the number of edges from Root to the deepest leaf.
- **Height of Root == Height of Tree == Depth of Tree == number of edges in the longest path of the tree from Root to a leaf.** → depth - 1
- **Size of a tree** : It is the total number of nodes present in the tree.
- **Binary Tree** : A tree in which each node can have at most two child nodes. Here each node contains one data part and two link parts.
- **Complete Binary Tree** : It is a binary tree in which every node has an out-degree of either 2 or 0.
- **Binary Search Tree** : A binary tree in which the value of the left child is smaller than the value of the parent and the value of the right child is larger than the value of the parent node.

**Example :**

**Given a binary tree :**



**Answer the following questions on the given binary tree [solved] :**

- Which is the Root? H
- How many leaf nodes are present and which are they?  
5 leaf nodes and they are B , C , G , L , R
- Which is the parent node of P? S
- Which are the child node/nodes of D and A?  
child node of D is A child nodes of A are B , C

- (v) What is the In-degree and Out-degree of R?  
In-degree of R = 1 and Out-degree of R = 0
- (vi) What is the height of the tree? 4
- (vii) Write the path from node H to G. Path : H S P F G
- (viii) Which is the sibling node of L? R
- (ix) What is the In-Degree and Out-degree of H?  
In-degree of H = 0 and Out-degree of H = 2
- (x) What is the height of node A and node S?  
Height of node A is 1 and height of node S is 3.