

Introduction To Binary Trees

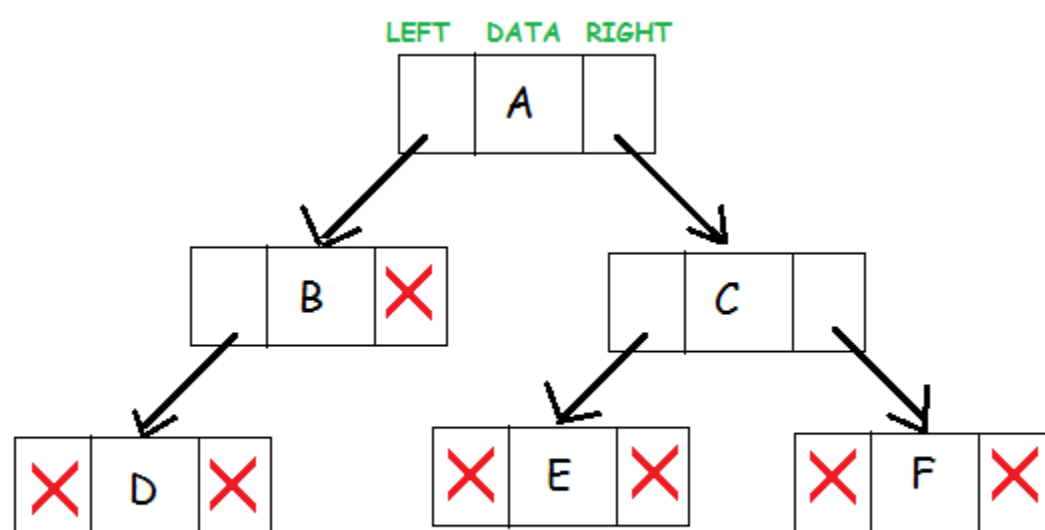
A binary tree is a hierarchical data structure in which each node has at most two children generally referred as left child and right child.

Each node contains three components:

1. Pointer to left subtree
2. Pointer to right subtree
3. Data element

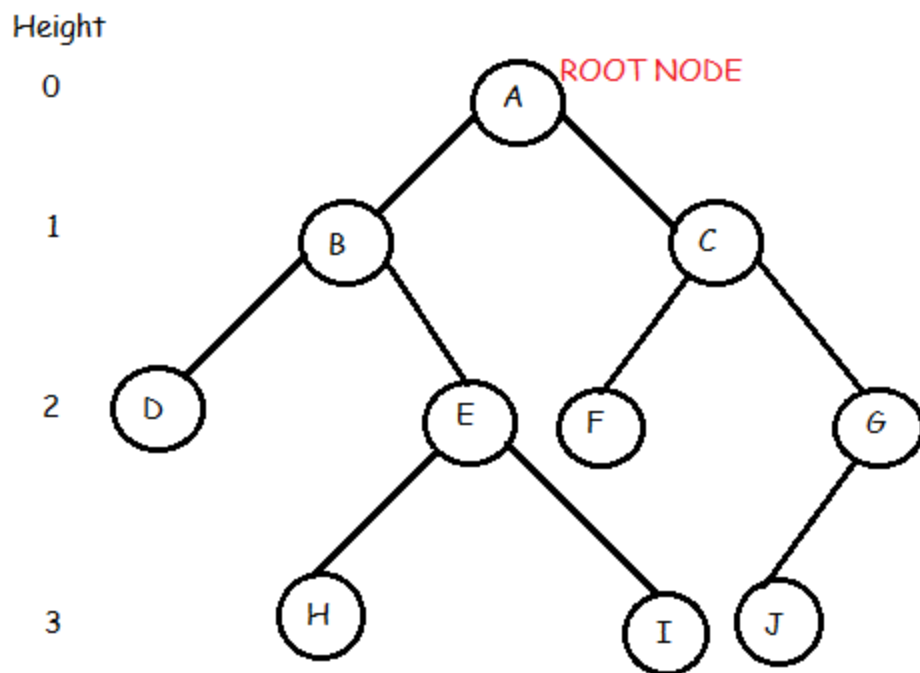
The topmost node in the tree is called the root. An empty tree is represented by **NULL** pointer.

A representation of binary tree is shown:



Binary Tree: Common Terminologies

- **Root:** Topmost node in a tree.
- **Parent:** Every node (excluding a root) in a tree is connected by a directed edge from exactly one other node. This node is called a parent.
- **Child:** A node directly connected to another node when moving away from the root.
- **Leaf/External node:** Node with no children.
- **Internal node:** Node with atleast one children.
- **Depth of a node:** Number of edges from root to the node.
- **Height of a node:** Number of edges from the node to the deepest leaf. Height of the tree is the height of the root.



In the above binary tree we see that root node is **A**. The tree has 10 nodes with 5 internal nodes, i.e, **A,B,C,E,G** and 5 external nodes, i.e, **D,F,H,I,J**. The height of the tree is 3. **B** is the parent of **D** and **E** while **D** and **E** are children of **B**.

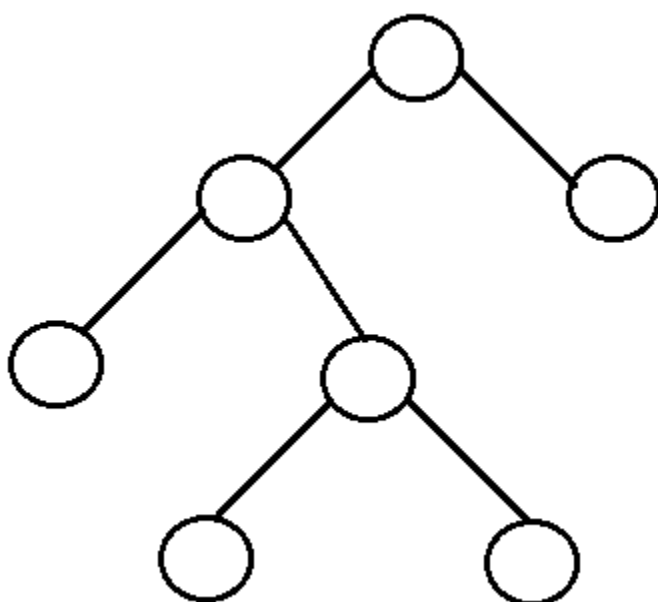
Advantages of Trees

Trees are so useful and frequently used, because they have some very serious advantages:

- Trees reflect structural relationships in the data.
- Trees are used to represent hierarchies.
- Trees provide an efficient insertion and searching.
- Trees are very flexible data, allowing to move subtrees around with minumum effort.

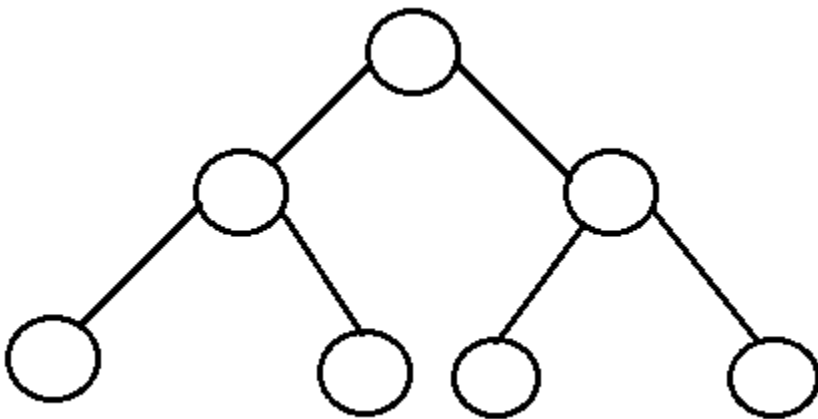
Types of Binary Trees (Based on Structure)

- **Rooted binary tree:** It has a root node and every node has atmost two children.
- **Full binary tree:** It is a tree in which every node in the tree has either 0 or 2 children.

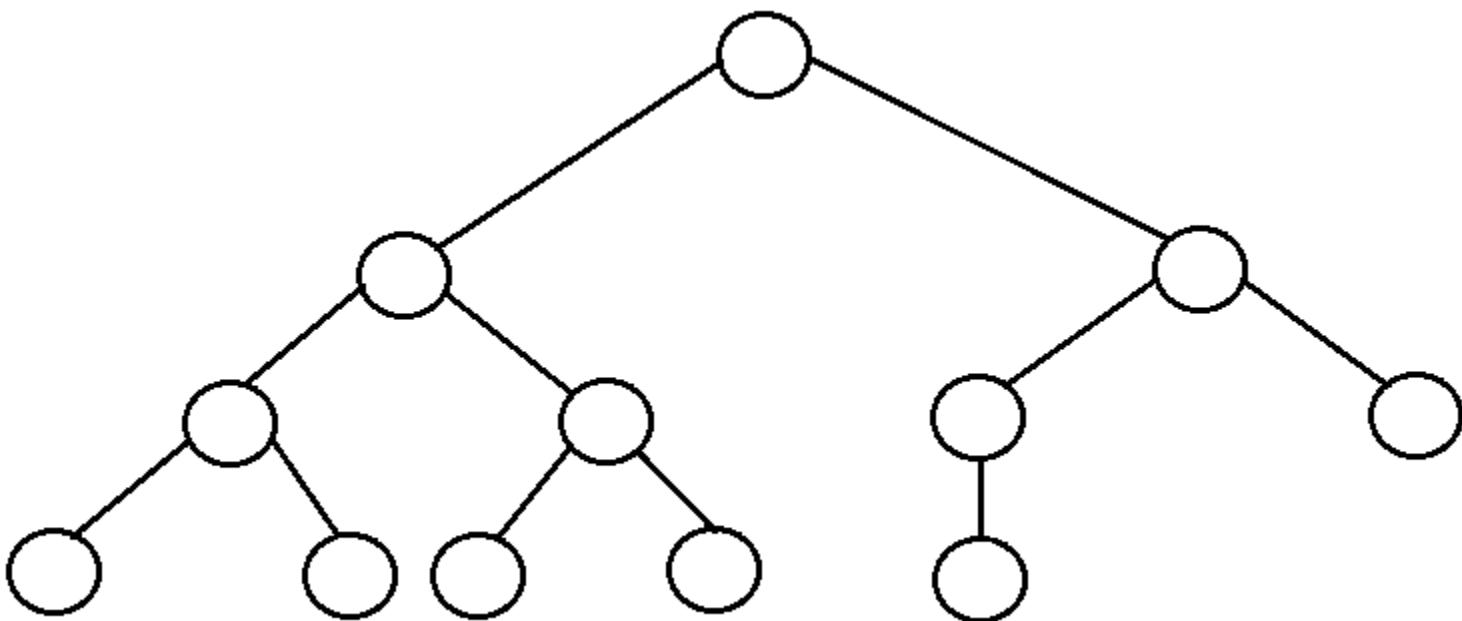


- The number of nodes, n , in a full binary tree is atleast $n = 2h - 1$, and atmost $n = 2^{h+1} - 1$, where h is the height of the tree.

- The number of leaf nodes l , in a full binary tree is number, L of internal nodes + 1, i.e, $l = L + 1$.
- **Perfect binary tree:** It is a binary tree in which all interior nodes have two children and all leaves have the same depth or same level.



- A perfect binary tree with l leaves has $n = 2l - 1$ nodes.
- In perfect full binary tree, $l = 2^h$ and $n = 2^{h+1} - 1$ where, n is number of nodes, h is height of tree and l is number of leaf nodes
- **Complete binary tree:** It is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible.



- The number of internal nodes in a complete binary tree of n nodes is $\text{floor}(n/2)$.
- **Balanced binary tree:** A binary tree is height balanced if it satisfies the following constraints:
 1. The left and right subtrees' heights differ by at most one, AND
 2. The left subtree is balanced, AND
 3. The right subtree is balanced

An empty tree is height balanced.

