Department of Computer Science and Engineering

PES UNIVERSITY

UE19CS202: Data Structures and its Applications (4-0-0-4-4)

Trees

Binary Trees: Basic Concept and Definitions

Dr. Shylaja S S
Ms. Kusuma K V

Binary Tree: is a Non Linear Data Structure

Definition: Finite set of elements that is either empty or is partitioned into three subsets

- First subset: is a single element, called the root
- Second subset: is a binary tree, called the left binary tree
- Third subset: is a binary tree, called the right binary tree

Figure 3 shows a Binary Tree with A as its root. Tree with nodes B, D, E, G form the left subtree and the tree with nodes C, F, H, I form the right subtree.

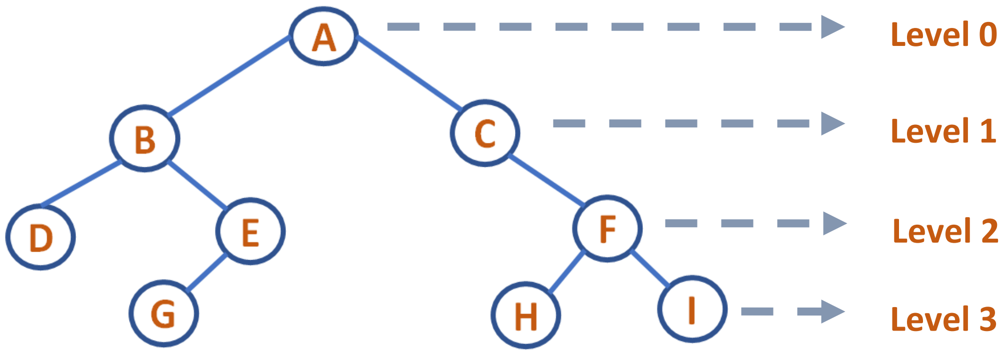


Figure 3: Binary Tree

Each element of a binary tree is called a **node** of the tree.
Left node B of A is called left child of A. Right node C of A is called the right child of A.
Two nodes are called **siblings** if they are left and right children of the same parent. A is called the parent of B and C. B and C are called siblings
A node which has no child is called **leaf node/external node**
A node which has a child is called the non **leaf node/internal node**
A node N1 is called the **ancestor** of a node N2 if N1 is either the parent of N2 or N1 is the parent of some ancestor of N2. A node N2 becomes the **descendent** of node N1. Descendent can be either the left descendent or the right descendent.
Level of a node: Root has Level 0; Level of any other node is one more than its parent.
Depth of a tree: Maximum level of any leaf in the tree (path length from the deepest leaf to the root)
Depth of a node: Path length from the node to the root
Height of a tree: Path length from the root node to the deepest leaf
Height of a node: Path length from the node to the deepest leaf

In Figure 3, D, G, H and I are leaf nodes. A, B, C, E and F are internal nodes. A is the ancestor of all the nodes in the tree. B is the left descendant of A. C is the right descendent of A.
Level of node A: 0, Level of node B, C: 1, Level of node D, E, F: 2, Level of node G, H, I: 3
Depth of tree: 3
Depth of node A: 0, Depth of node B, C: 1, Depth of node D, E, F: 2, Depth of node G, H, I: 3
Height of tree: 3
Height of Node A: 3, Height of Node B, C: 2, Height of Node E, F: 1, Height of Node D,G,H,I: 0

Figure 4 shows some structures that are not binary trees.

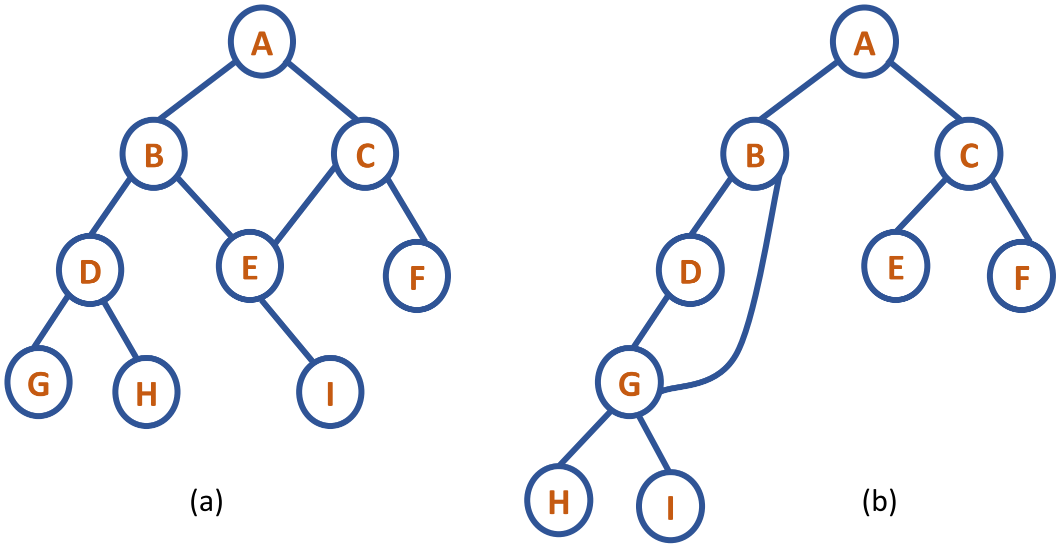


Figure 4: Structures that are not Binary Trees

Note: Different authors use the following tree terminologies in different ways.
Strictly Binary Tree: A Binary tree where every node has either zero/two children.

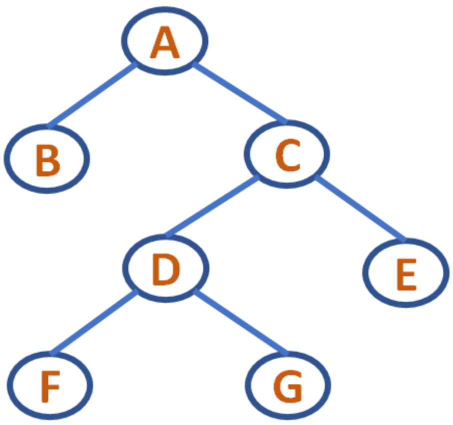


Figure 5: Strictly Binary Tree

Fully Binary Tree: A binary tree with all the leaves at the same level.

- If the binary tree has depth d, then there are 0 to d levels
- Total no. of nodes = $2^0+2^1+\ldots\ldots+2^d = 2^{(d+1)}-1$

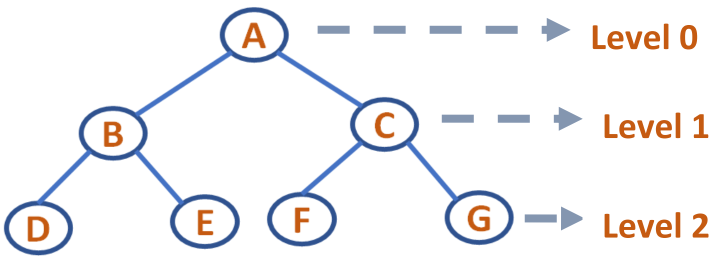


Figure 6: Full Binary Tree of depth 2

Complete Binary Tree: For a Complete Binary Tree with n nodes and depth d:

1. Any node nd at level less than d-1 has two children
2. For any node nd of the tree with a right descendent at level d, nd must have a left child and every left descendent of nd is either a leaf at level d or has two children

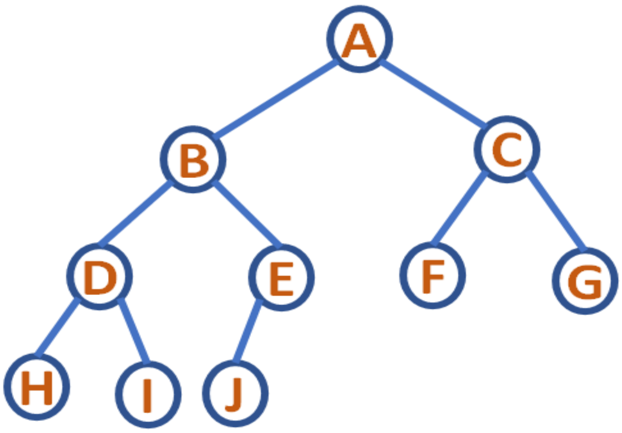
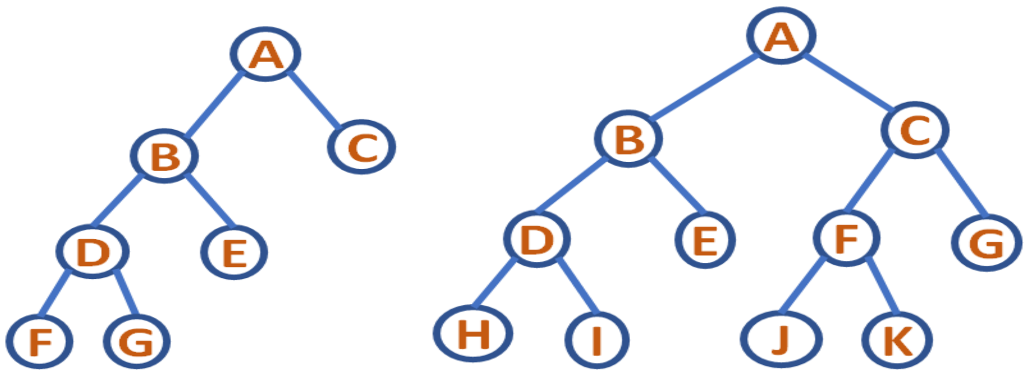


Figure 7: Complete Binary Tree

The binary tree of Figure 8(a) is not complete, since it contains leaves at levels 1, 2, and 3, thereby violating condition 1. The binary tree of Figure 8(b) satisfies condition1, since every leaf is either at level 2 or at level 3. However, condition 2 is violated, since A has a right descendent at level 3 (J) but also has a left descendant that is a leaf at level 2(E).



(a) (b)

Figure 8: Structures that are not Complete Binary Trees

Binary Tree Properties

- Every node except the root has exactly one parent
- A tree with n nodes has n-1 edges (every node except the root has an edge to its parent)
- A tree consisting of only root node has height of zero
- The total number of nodes in a full binary tree of depth d is $2^{(d+1)} - 1$, $d \geq 0$
- For any non-empty binary tree, if n_0 is the number of leaf nodes and n2 the nodes of degree 2, then $n_0 = n_2 + 1$