Data Structures & Algorithms

Tree

Tree

A tree is a data structure consisting of finite non-empty set of nodes organized as a hierarchy, with a designated root node (have no parent) and its children (zero or more) represented as a set of linked nodes, where each node is consisting of a value, together with a list of references to nodes (the "children") which are called subtrees of the root.

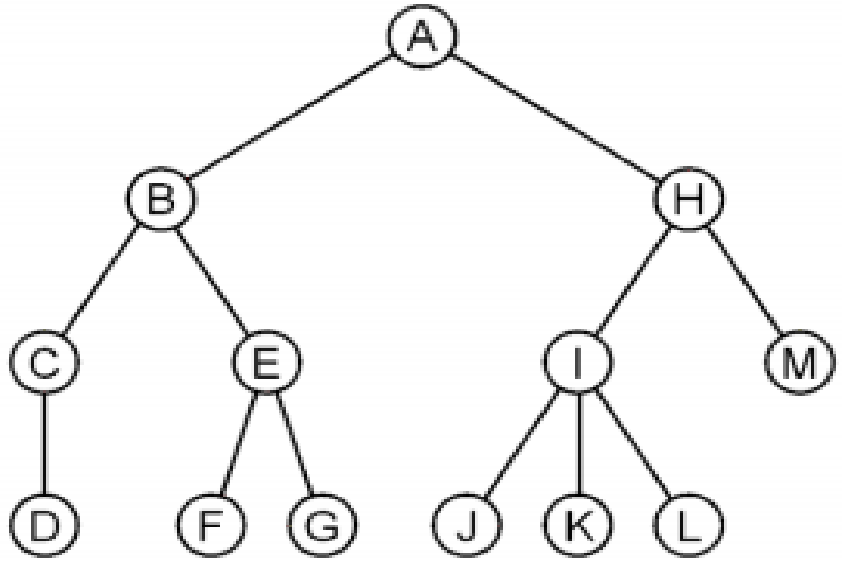


Figure A tree with a root storing the value 'A'.

Tree

Terminologies

Siblings : Two nodes that are children of the same parent.

The degree of a node : the number of children a particular node has.

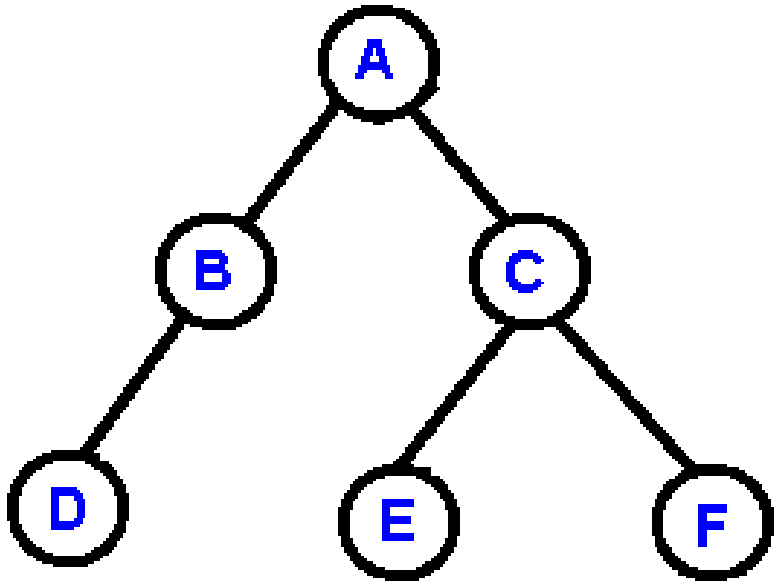
Leaf nodes/external node/terminal node : Nodes of degree zero. A node that has no child nodes.

Non leaf/internal nodes : All internal nodes which are not leaf nodes. An internal node has at least one child.

For example,
In the previous Figure, the degree of node E is 2, its children are F and G, and its parent is node B. The nodes D, F, G, J, K, L, and M are leaf nodes, while nodes A, B, H, C, E, and I are non leaf nodes.

Binary Tree

In a binary tree each internal node can have at most/maximum of two child nodes connected to it, i.e. all the nodes have maximum two children, which are referred to as the left child and the right child.



Binary Tree

Structure of binary tree nodes

```
typedef struct BinaryTreeNode
{
    struct BinaryTreeNode *left;   // Left child
    int data; // The data in the node
    struct BinaryTreeNode *right;    // Right child
} BTnode;
BTnode *root;
root = NULL;  // Empty tree
```

Create a node in a binary tree

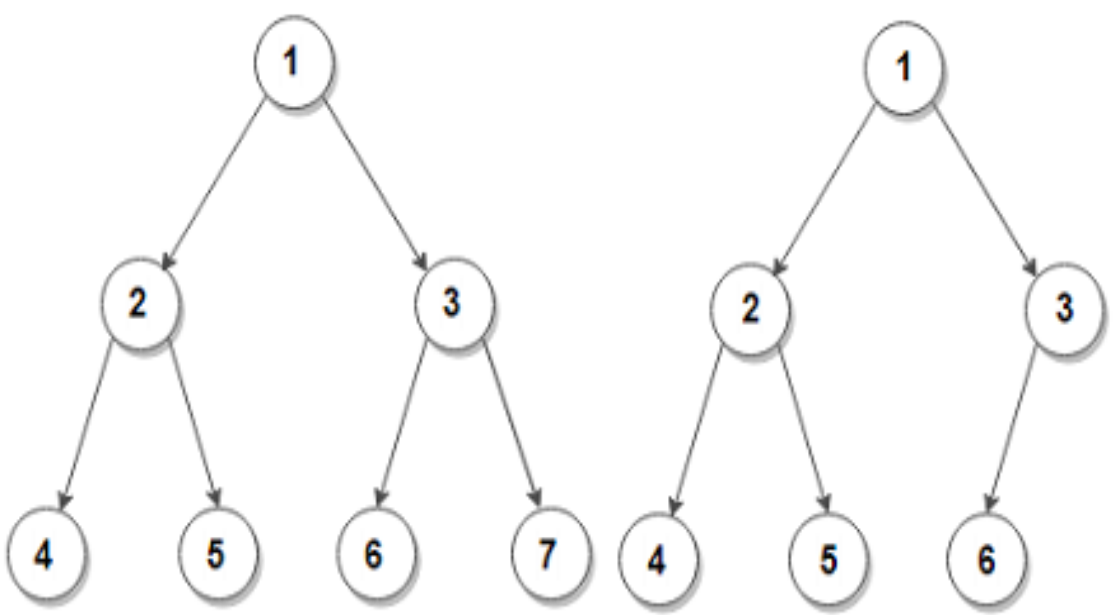
```
new_node = (BTnode *) malloc(sizeof(BTnode));
root ->data = item;
root ->left = NULL;
root ->right = NULL;
root = new_node;
```

Variations of Binary Tree

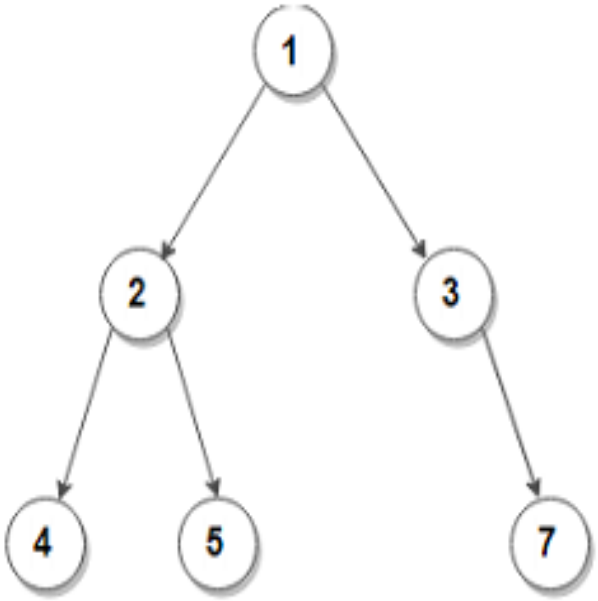
A full binary tree (a.k.a proper binary tree/2-tree)is a tree in which every node other than the leaves has 2 children.

In a **complete** binary tree every level, except possibly the last/bottom, is completely filled, and all nodes in the last level are as far left as possible (filled from left to right).

Full Binary Tree vs. Complete Binary Tree

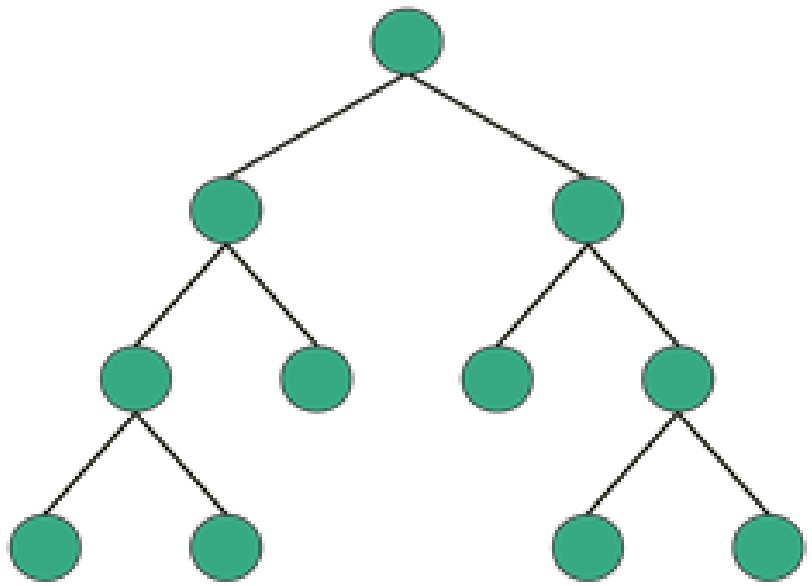


Complete Binary Tree

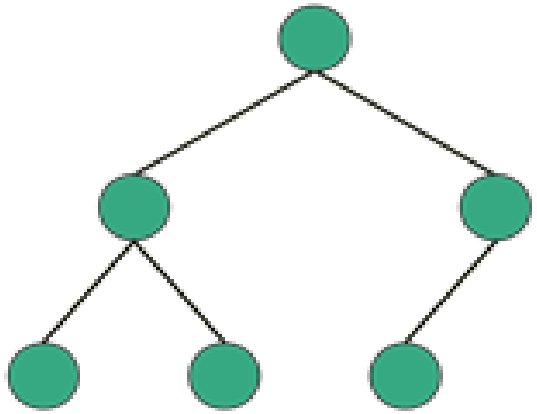


Not a Complete Binary Tree

Full Binary Tree vs Complete Binary Tree

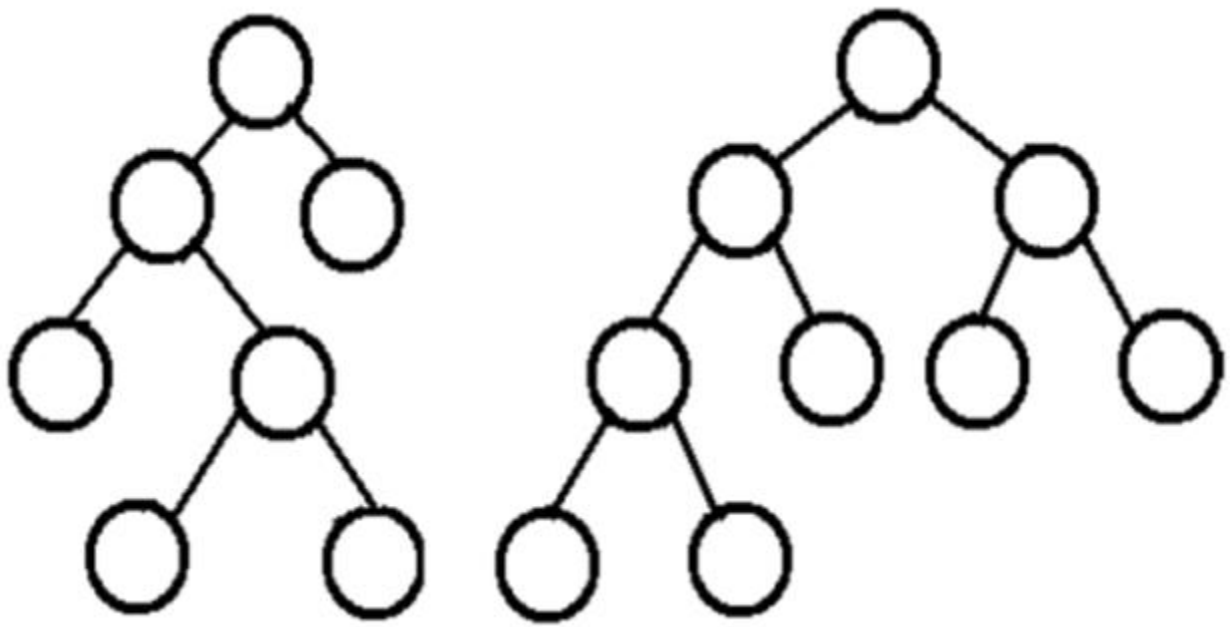


Full



Complete

Full Binary Tree vs Complete Binary Tree



Binary Tree

Representation/Implementation

Binary trees can be implemented in two ways:

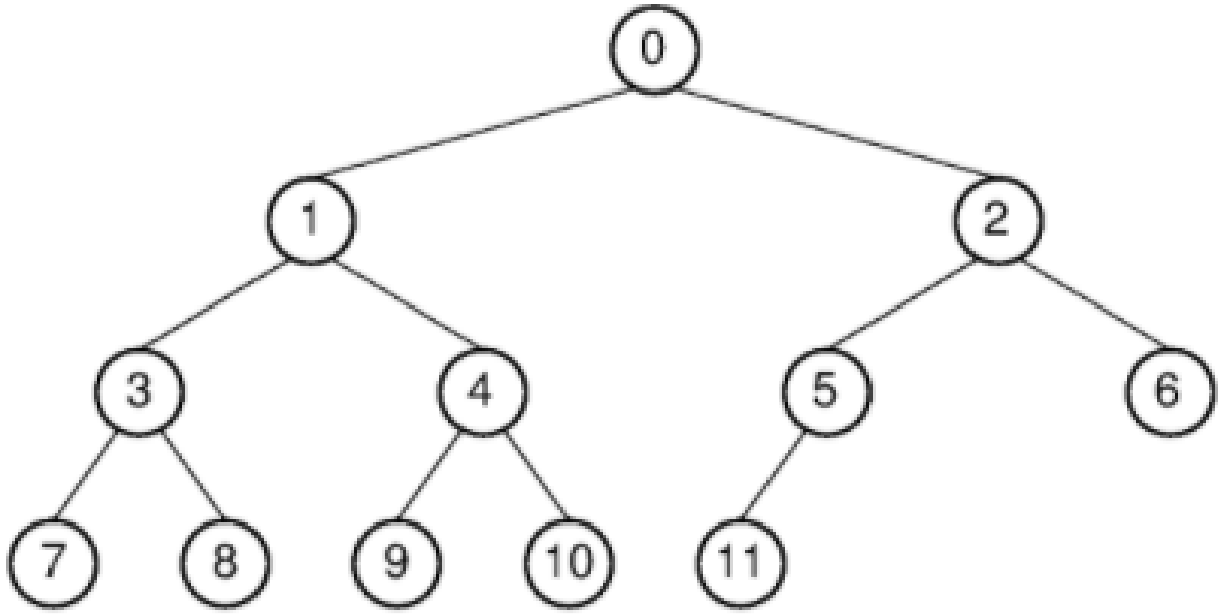
a) as one dimensional arrays and

b) as doubly linked list

Binary Tree

An Array Representation of binary tree

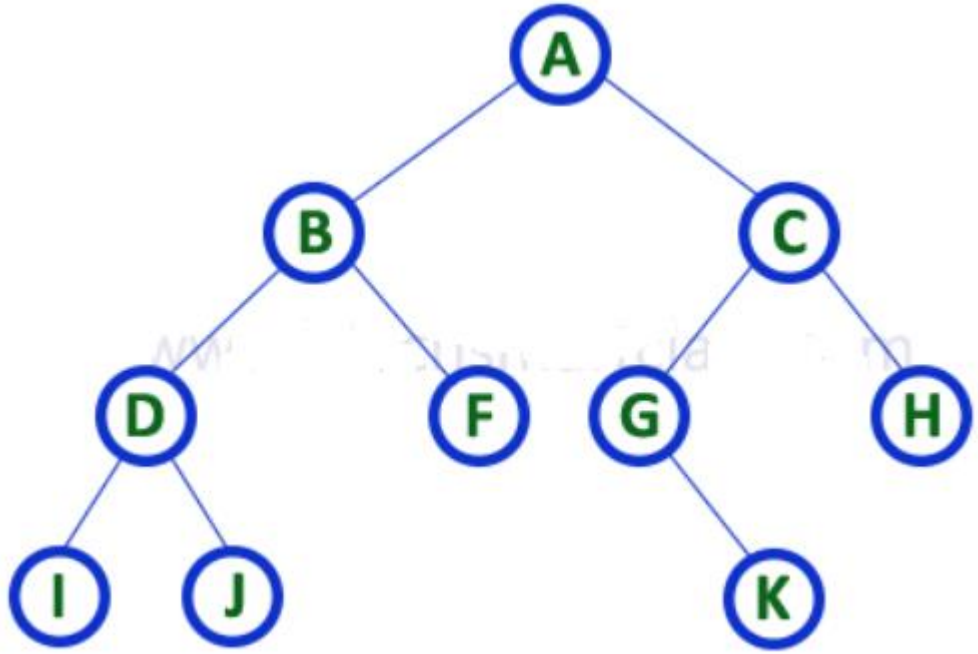
The left child of the node at index (i) is at index left(i) = 2i+1 and the right child of the node at index is at index right(i) = 2i+2. The parent of the node at index i is at index parent(i) = (i-1)/2.



Position	0	1	2	3	4	5	6	7	8	9	10	11
Parent	--	0	0	1	1	2	2	3	3	4	4	5
Left Child	1	3	5	7	9	11	--	--	--	--	--	--
Right Child	2	4	6	8	10	--	--	--	--	--	--	--

Binary Tree

Represent the following binary tree using an array



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
A	B	C	D	F	G	H	I	J	.	.	.	K	.	.	.	.	.	.	.	.	.	.

Binary Tree

A linked list Representation of binary tree

The previous example of the binary tree represented using Linked list is shown as follows