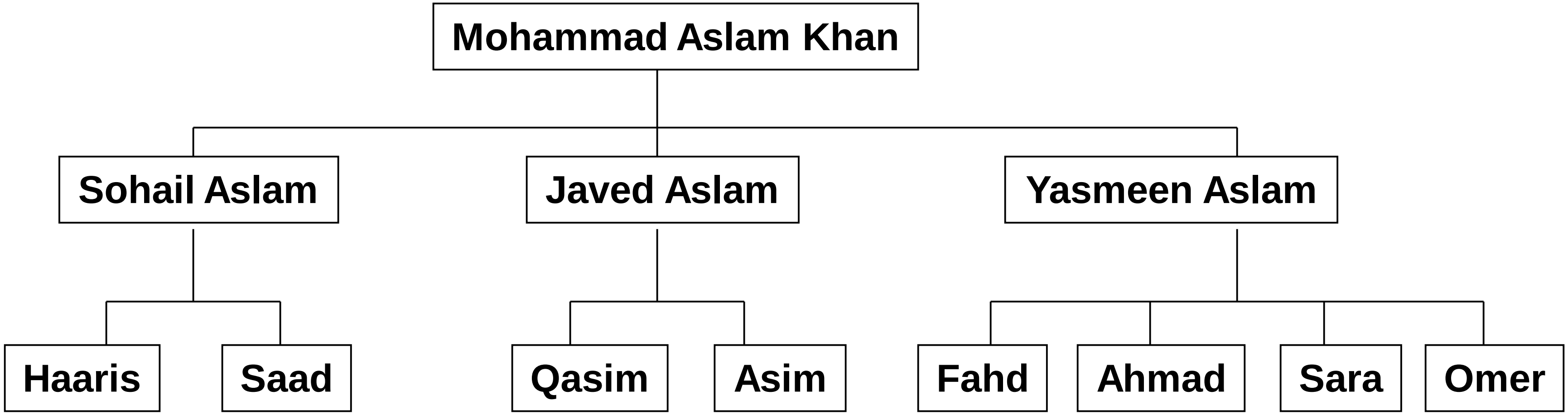
Tree Data Structure

Introduction

Tree Data Structures

- There are a number of applications where linear data structures are not appropriate.
- Consider a genealogy tree of a family.



Mohammad Aslam Khan

Sohail Aslam

Javed Aslam

Yasmeen Aslam

Haaris

Saad

Qasim

Asim

Fahd

Ahmad

Sara

Omer

Tree Data Structure

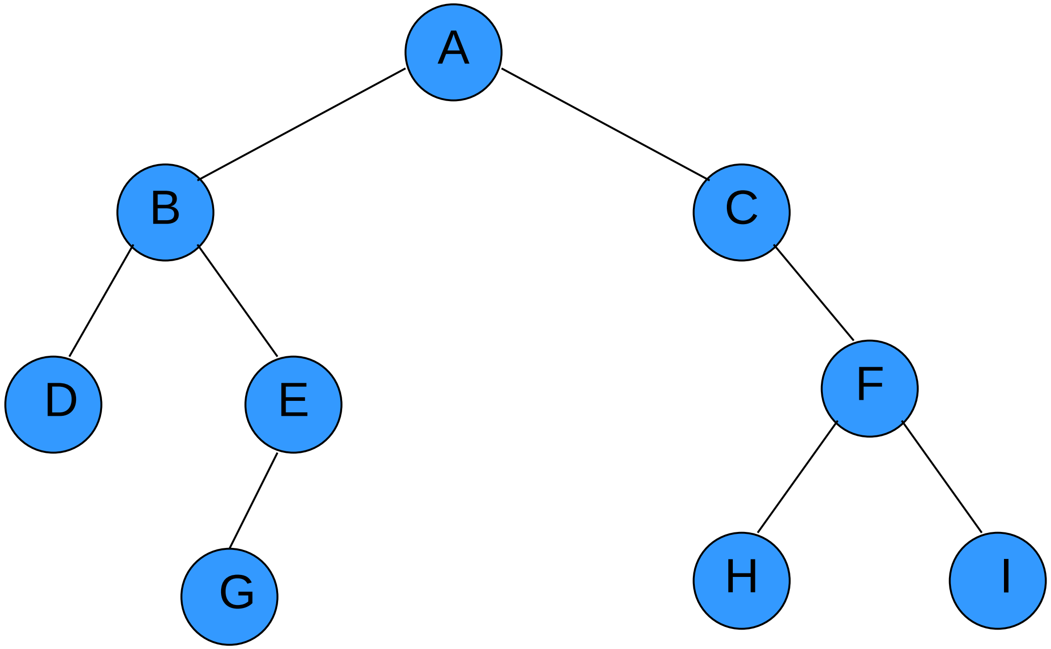
- A linear linked list will not be able to capture the tree-like relationship with ease.
- Shortly, we will see that for applications that require searching, linear data structures are not suitable.
- We will focus our attention on *binary trees*.

Binary Tree

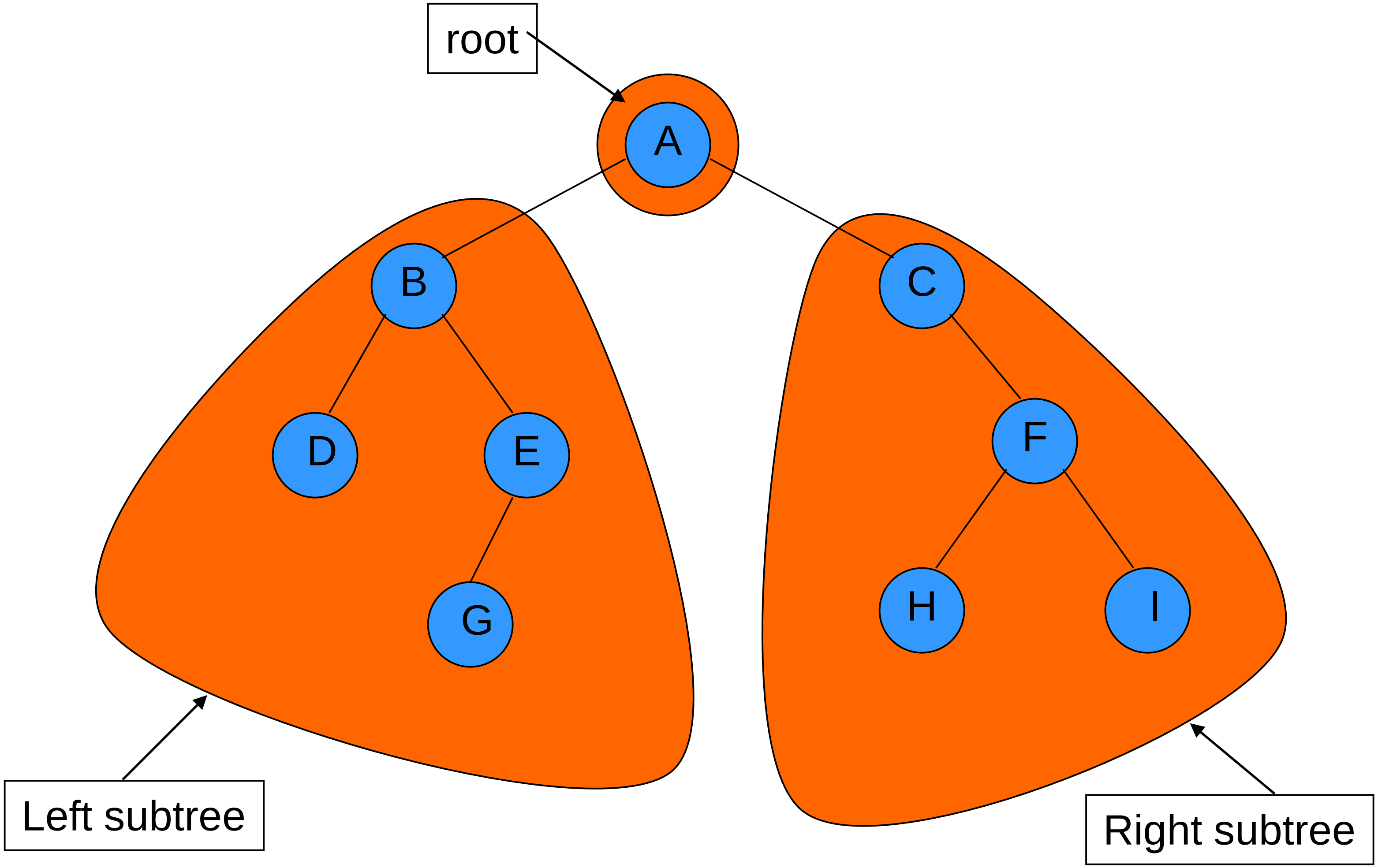
- A *binary tree* is a finite set of elements that is either empty or is partitioned into *three* disjoint subsets.
- The first subset contains a single element called the *root* of the tree.
- The other two subsets are themselves binary trees called the *left* and *right subtrees*.
- Each element of a binary tree is called a *node* of the tree.

Binary Tree

- Binary tree with 9 nodes.

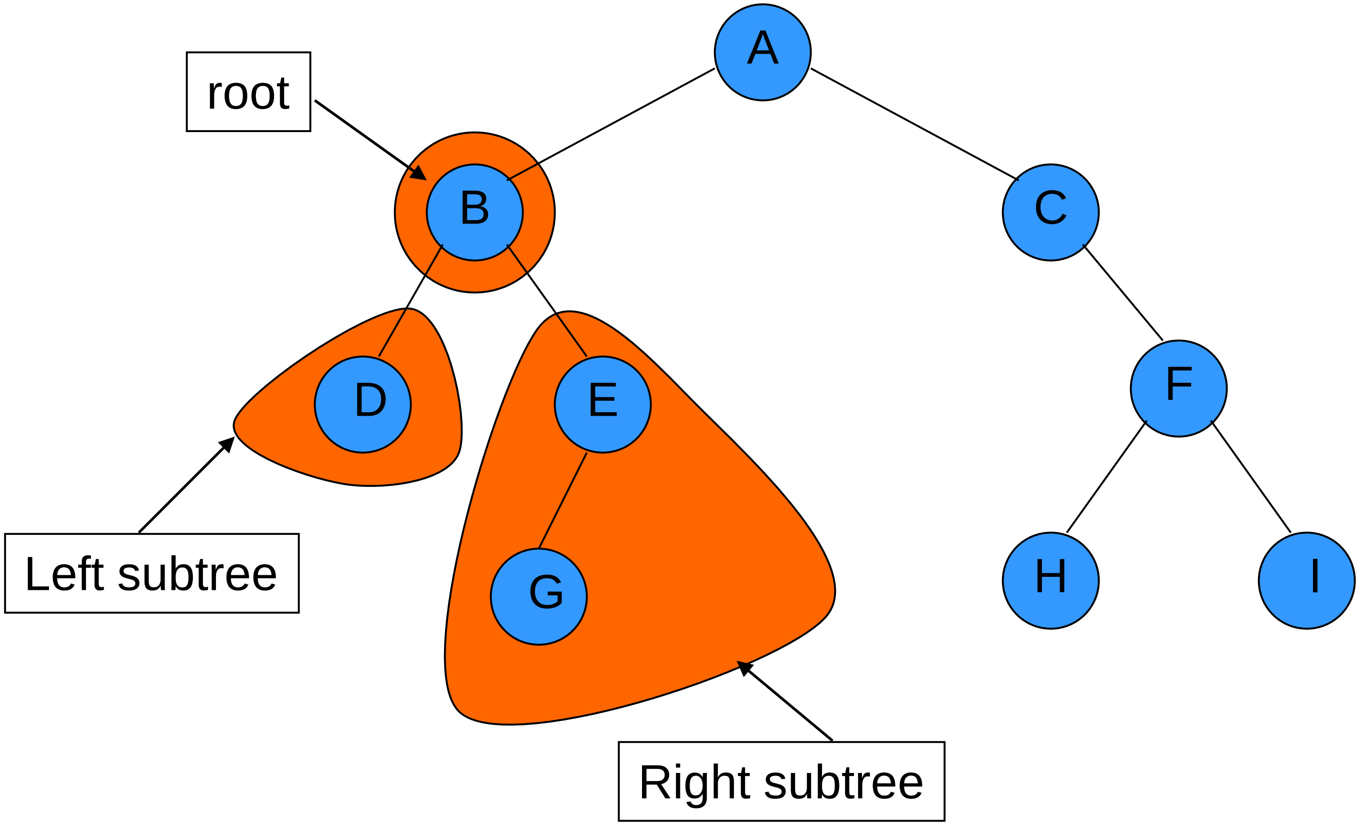


Binary Tree



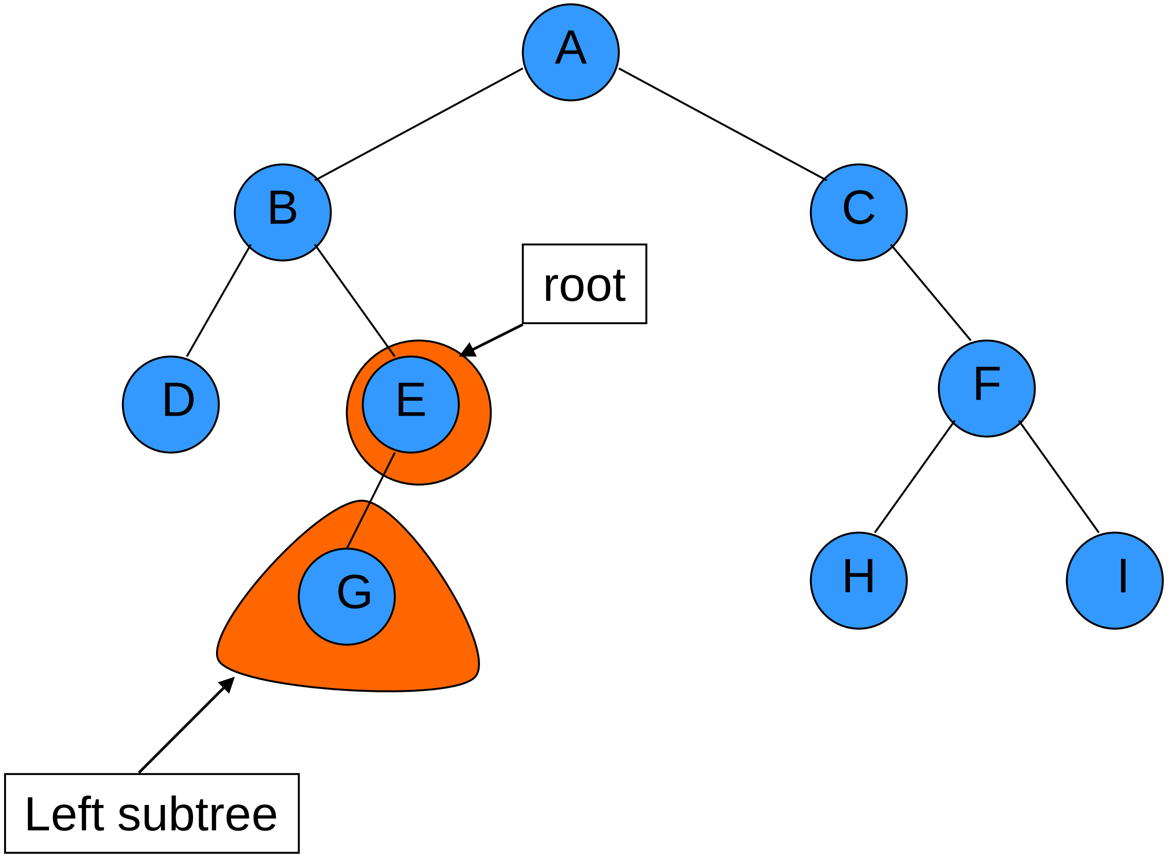
Binary Tree

- Recursive definition



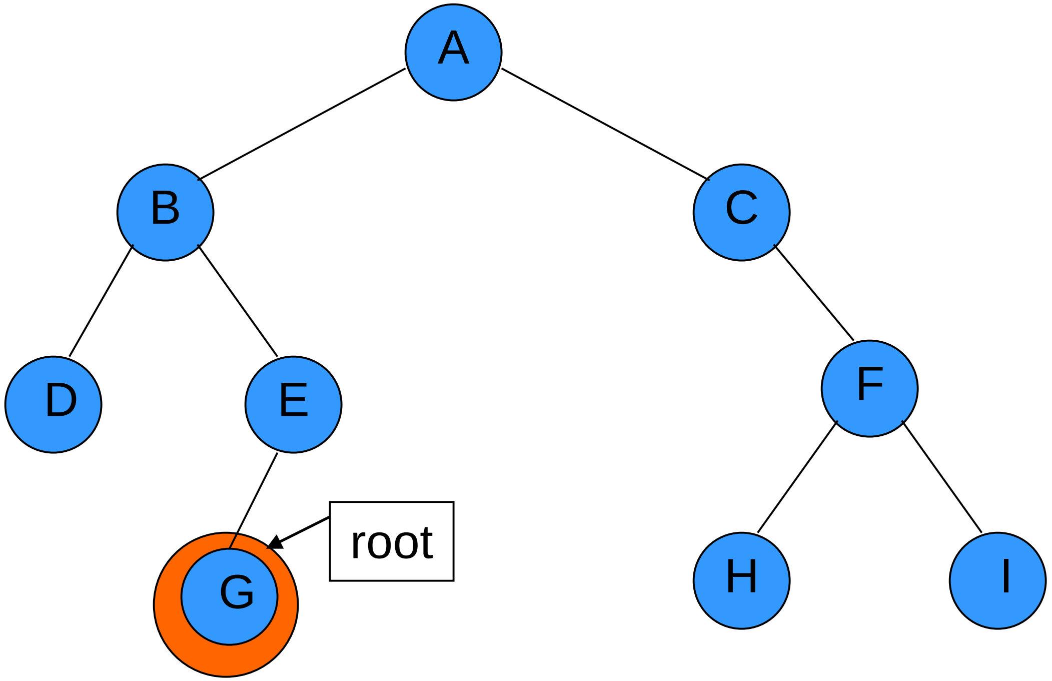
Binary Tree

- Recursive definition



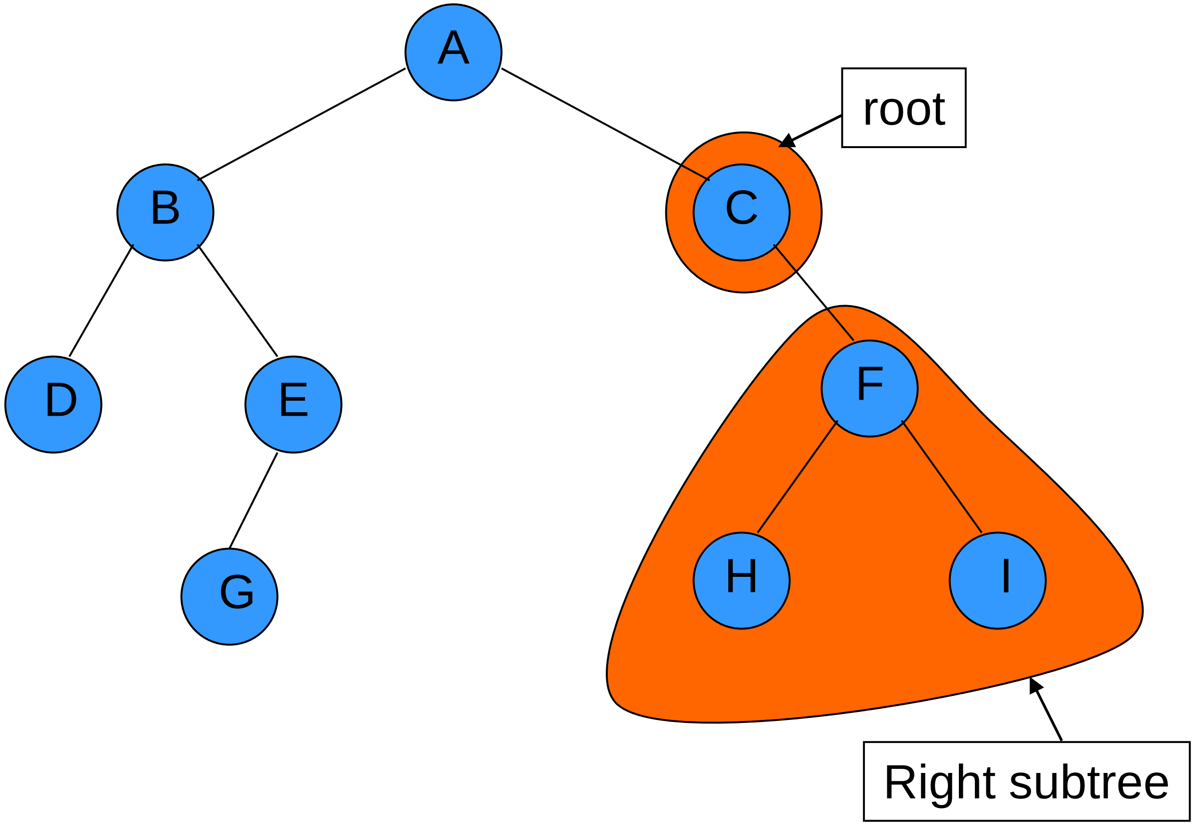
Binary Tree

- Recursive definition



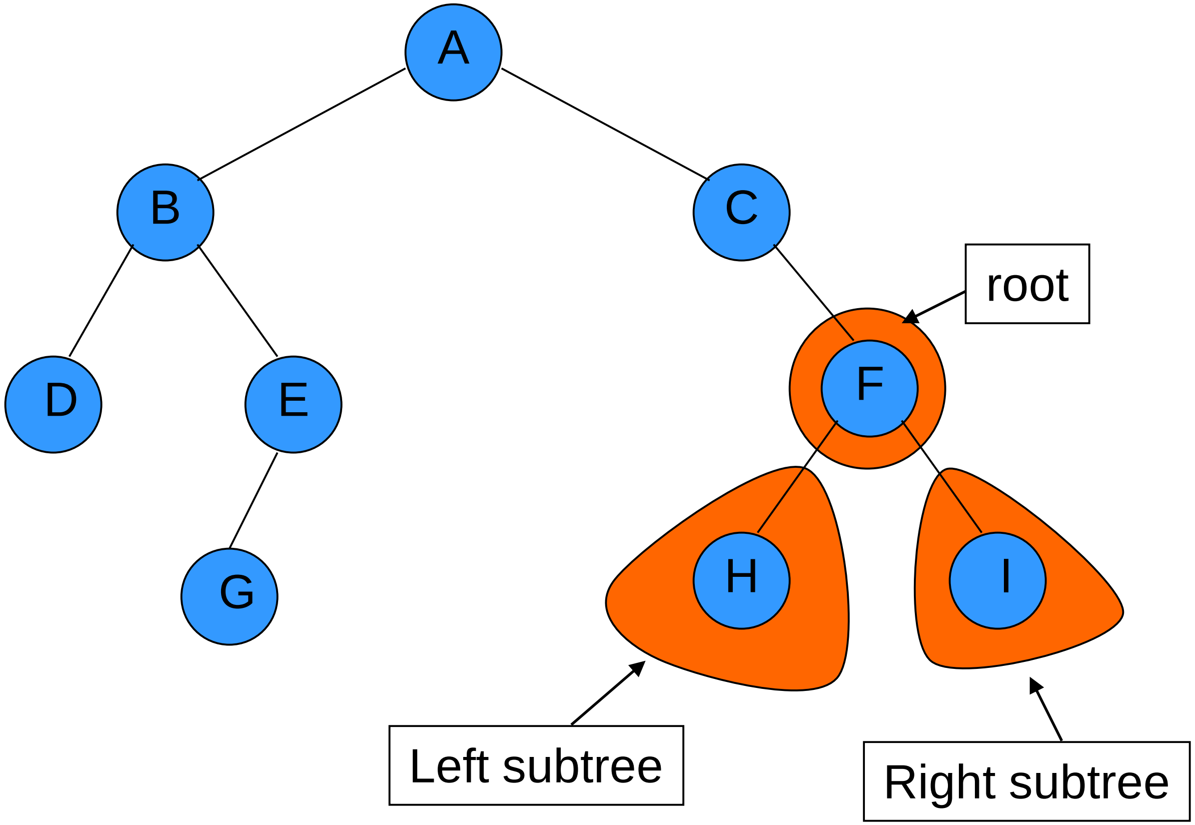
Binary Tree

- Recursive definition



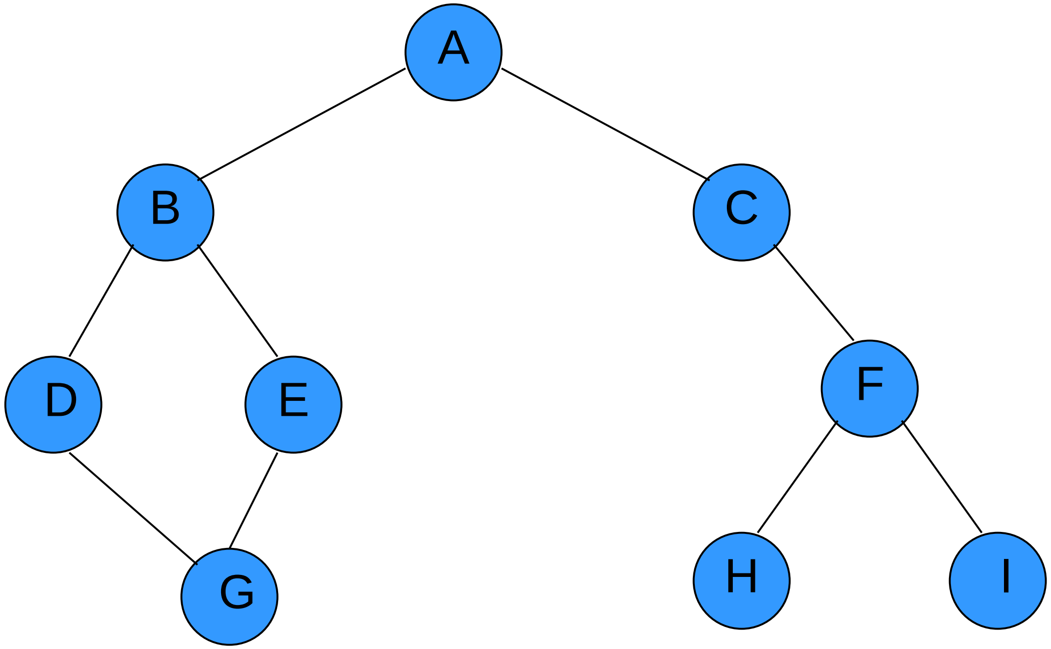
Binary Tree

- Recursive definition



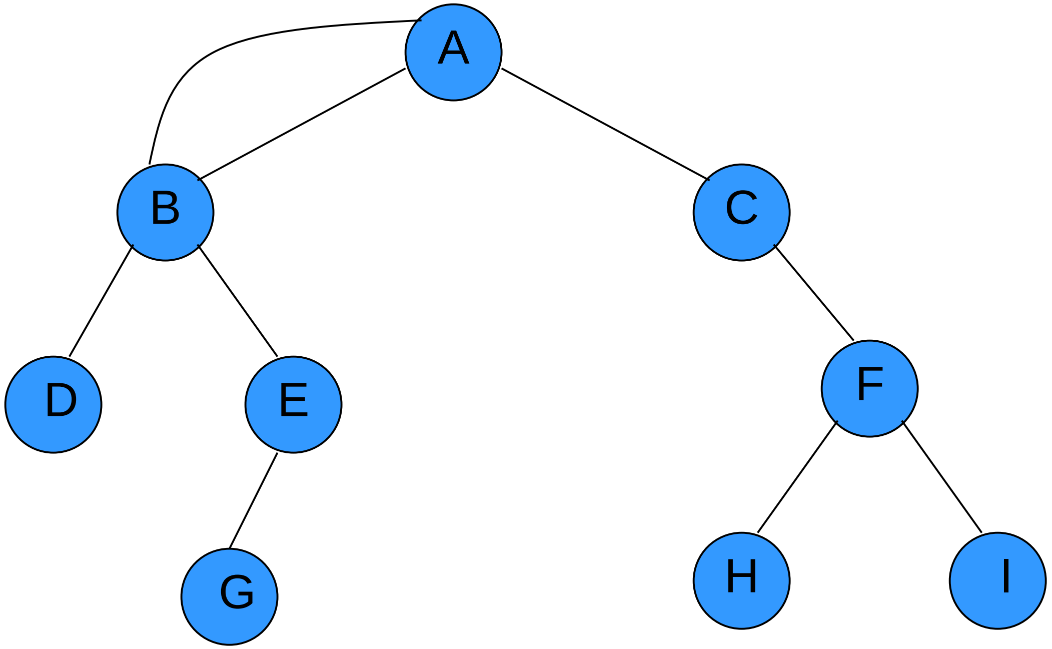
Not a Tree

- Structures that are not trees.



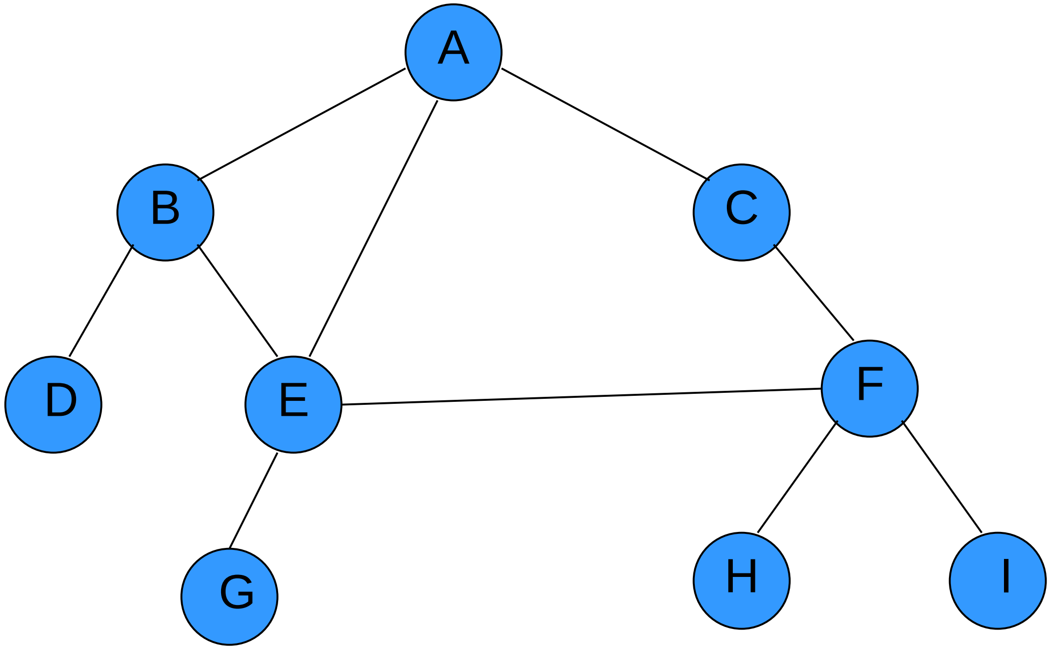
Not a Tree

- Structures that are not trees.

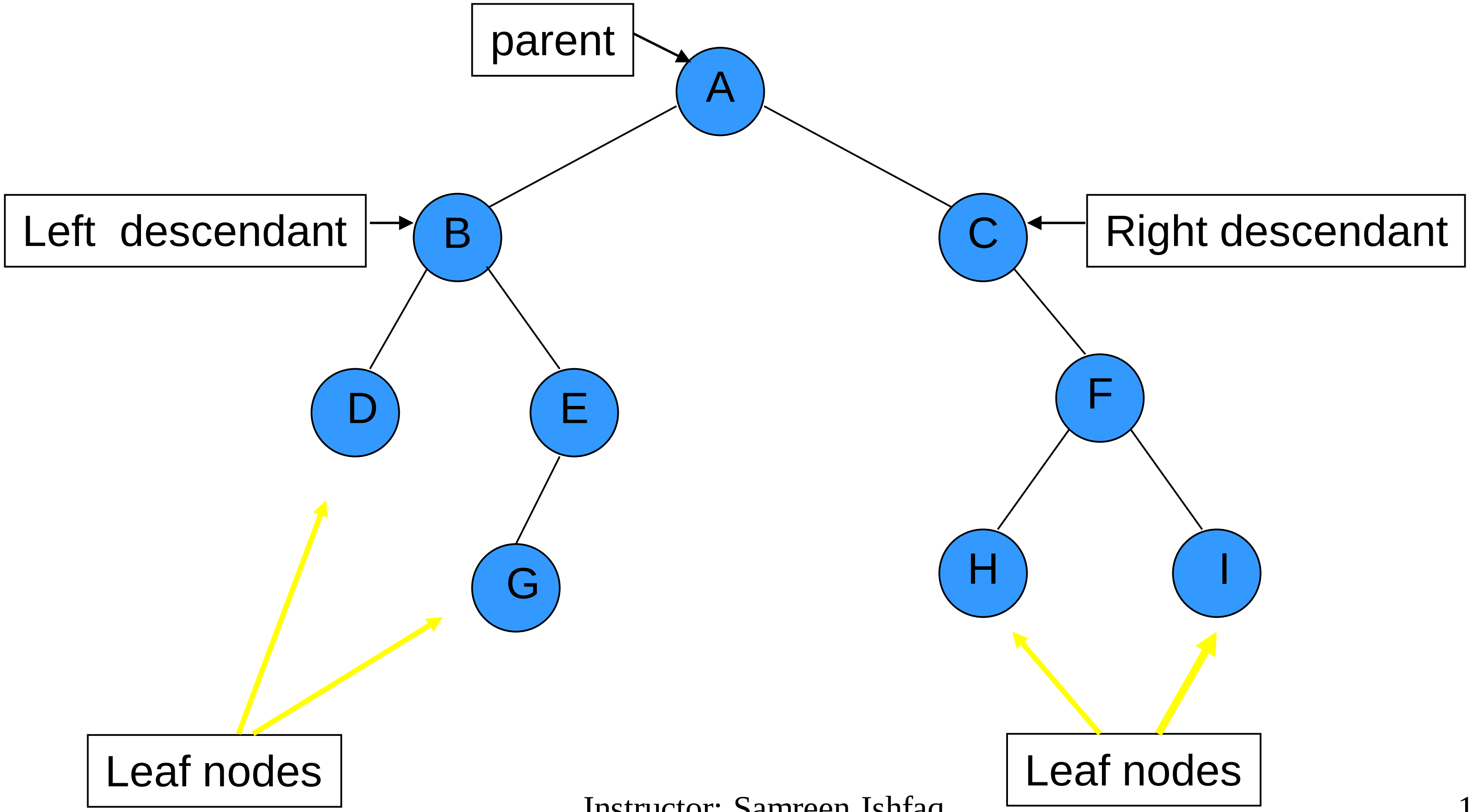


Not a Tree

- Structures that are not trees.

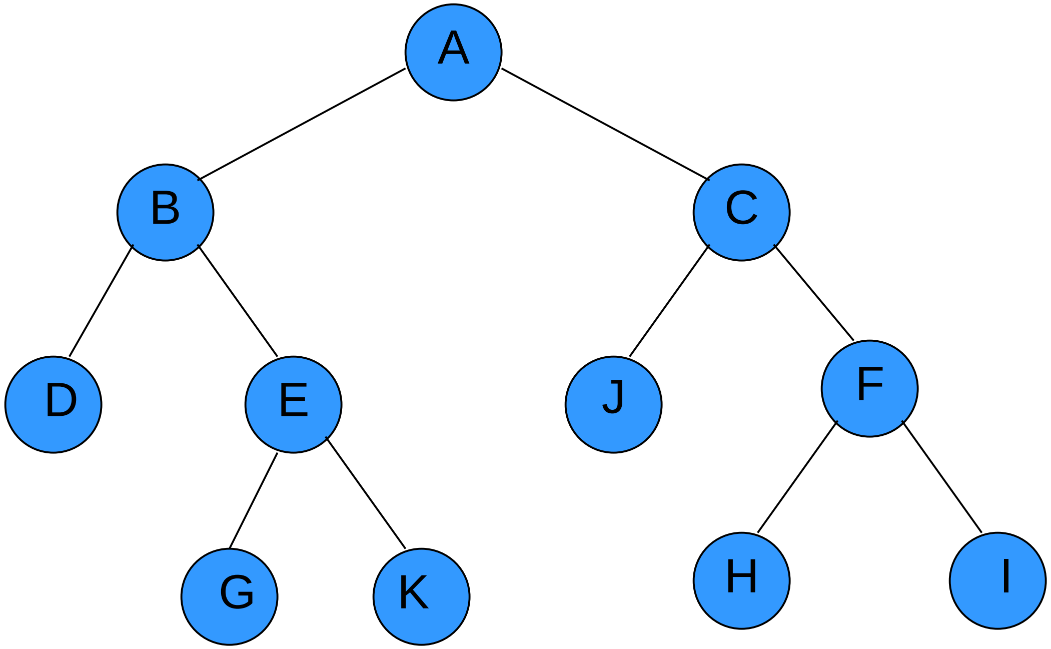


Binary Tree: Terminology



Binary Tree

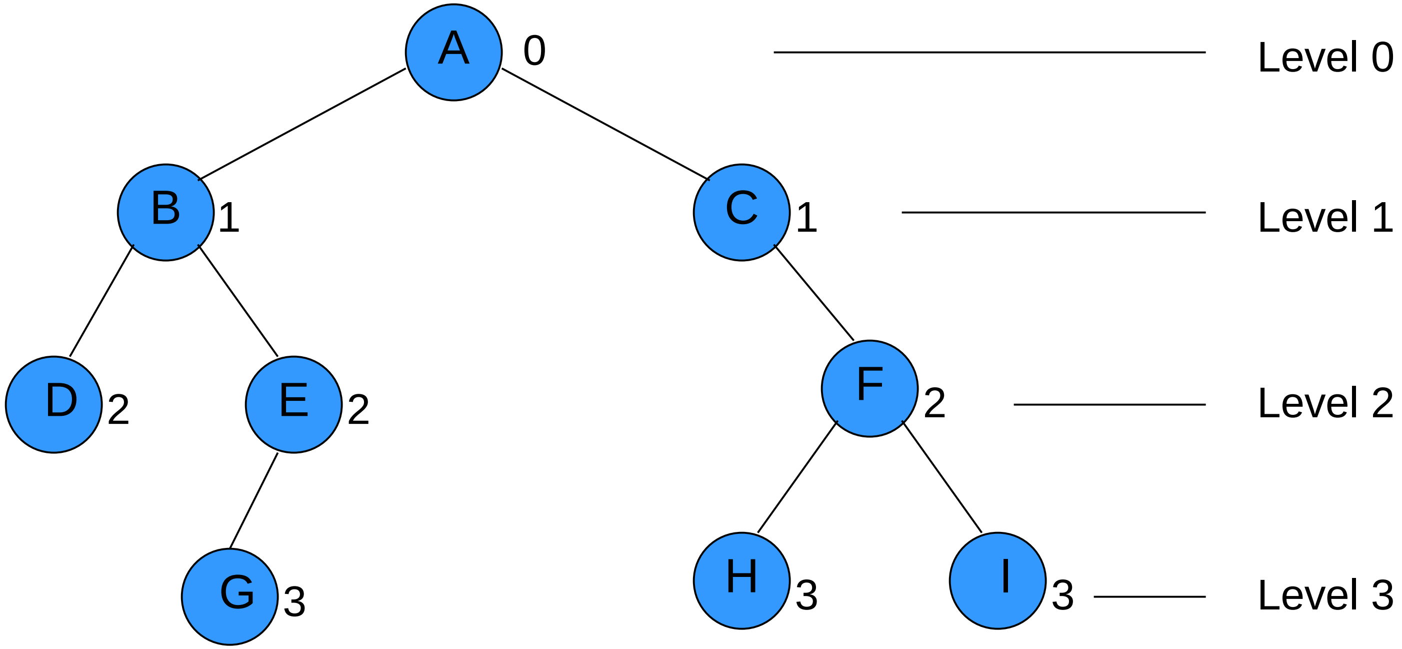
- If every non-leaf node in a binary tree has non-empty left and right subtrees, the tree is termed a *strictly binary tree*.



Level of a Binary Tree Node

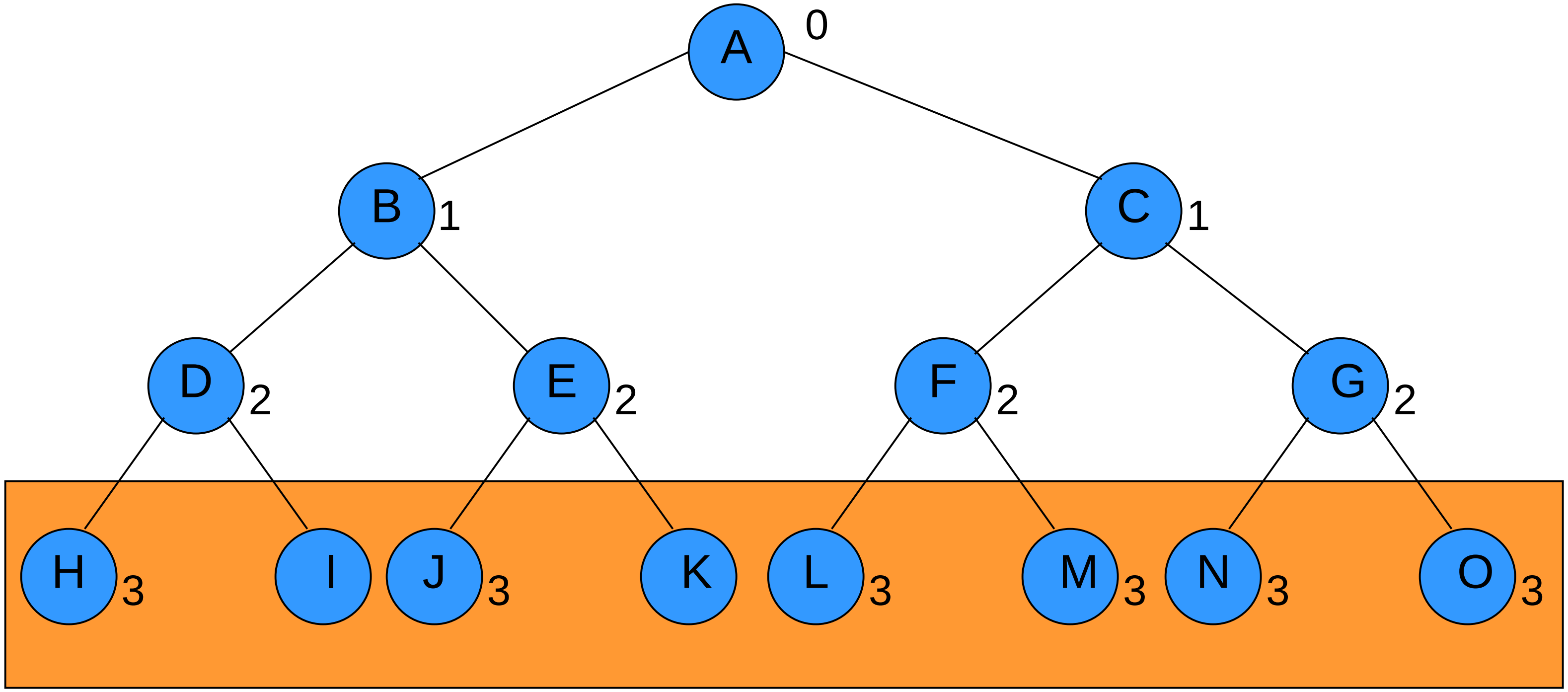
- The *level* of a node in a binary tree is defined as follows:
  - Root has level 0,
  - Level of any other node is one more than the level its parent (father).
- The *depth* of a binary tree is the maximum level of any leaf in the tree.

Level of a Binary Tree Node

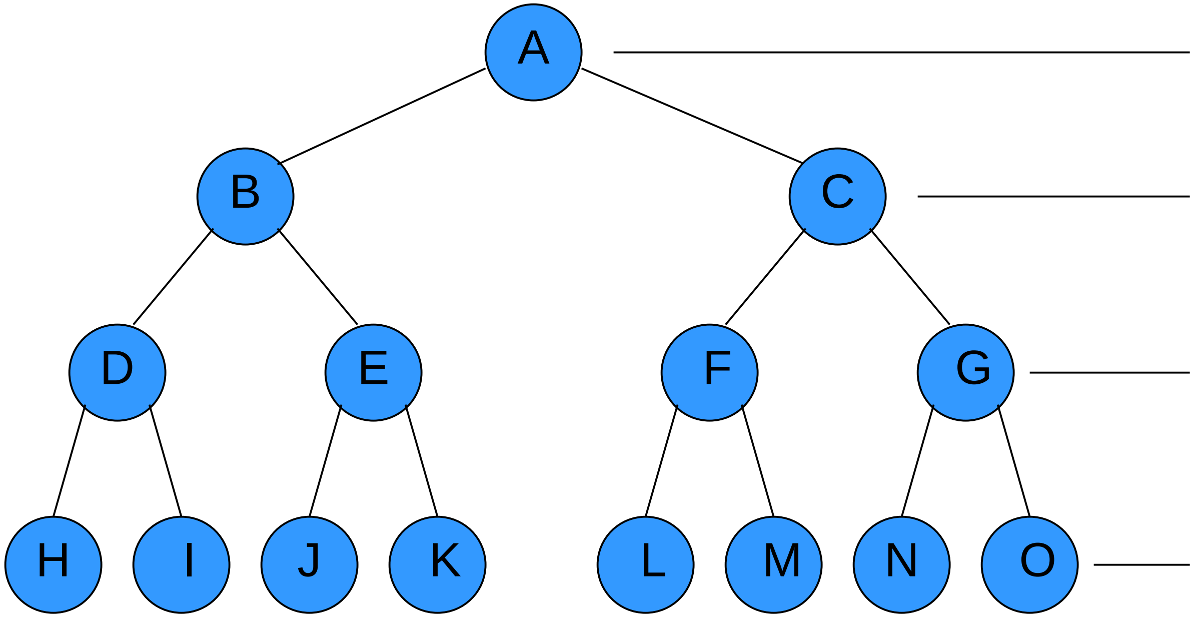


Complete Binary Tree

- A *complete binary tree* of depth *d* is the strictly binary all of whose leaves are at level *d*.



Complete Binary Tree



Level 0: 2^0 nodes

Level 1: 2^1 nodes

Level 2: 2^2 nodes

Level 3: 2^3 nodes

Complete Binary Tree

$$\sum_{i=1}^{n} r^i = \frac{r^{n+1} - 1}{r - 1}$$

- At level *k*, there are 2^k nodes.
- Total number of nodes in the tree of depth *d*:

$$2^0 + 2^1 + 2^2 + \ldots\ldots\ldots\ldots + 2^d = \overset{d}{\underset{j=0}{\textcircled{1}}}\ 2^j = 2^{d+1} - 1$$

- In a complete binary tree, there are 2^d leaf nodes and $(2^d - 1)$ non-leaf (inner) nodes.

Total number of nodes in a tree of depth 4

- $2^{d+1} - 1 = 2^{4+1} - 1$

  $= 2^5 - 1$

  $= 32 - 1$

  $= 31$ number of nodes

Complete Binary Tree

- If the tree is built out of ‘n’ nodes then

$$n = 2^{d+1} - 1$$

$$\text{or} \quad \log_2(n+1) = d+1$$

$$\text{or} \quad d = \log_2(n+1) - 1$$

- I.e., the depth of the complete binary tree built using ‘n’ nodes will be $\log_2(n+1) - 1$.
- For example, for n=100,000, $\log_2(100001)$ is less than 20; the tree would be 20 levels deep.
- The significance of this shallowness will become evident later.