DATA STRUCTURES

LECTURE-11

TREE

Dr. Sumitra Kisan

Tree

Tree data structure is a hierarchical structure that is used to represent and organize data in a way that is easy to navigate and search. It is a collection of nodes that are connected by edges and has a hierarchical relationship between the nodes.

A data structure which consists of

- a finite set of elements called nodes or vertices
- a finite set of directed arcs which connect the nodes

If the tree is nonempty

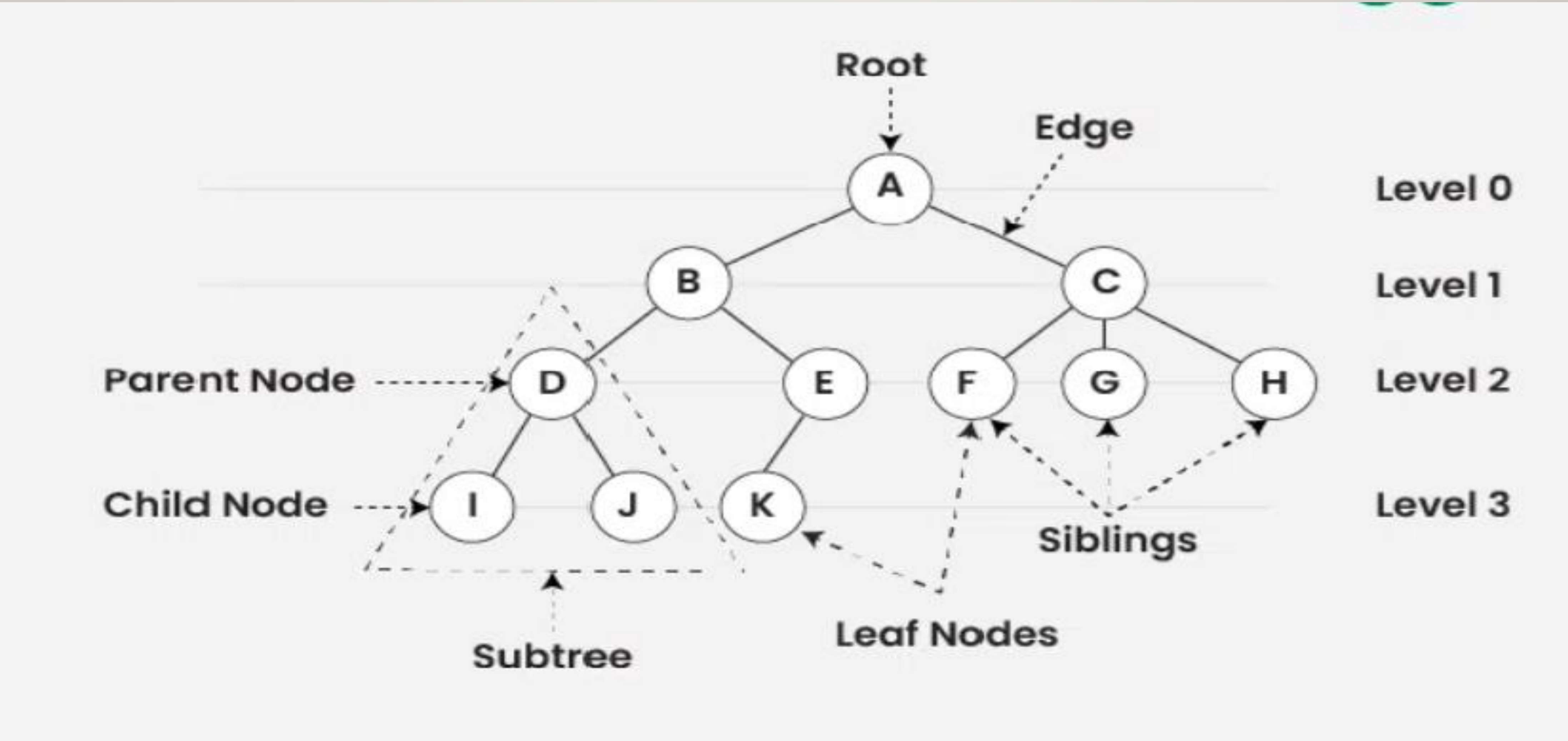
- one of the nodes (the root) has no incoming arc
- every other node can be reached by following a unique sequence of consecutive arcs

Tree is a non-linear data structure.

Basic Terminologies In Tree Data Structure

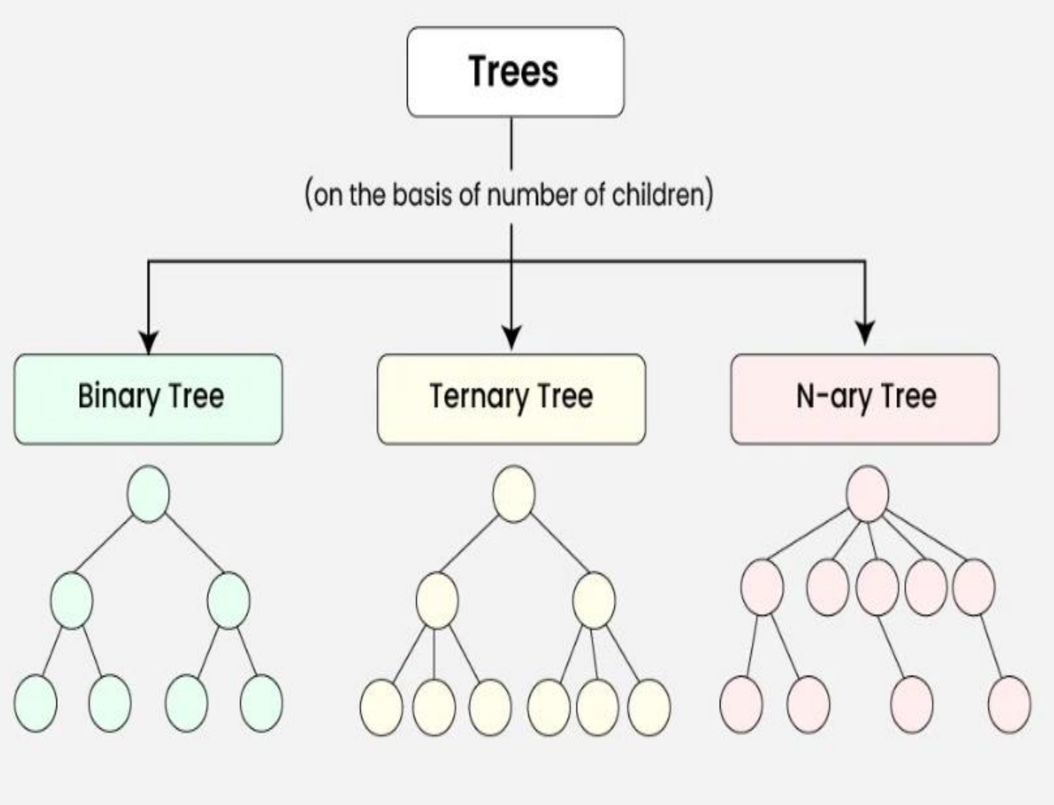
- **Parent Node:** The node which is a predecessor of a node is called the parent node of that node. **{B}** is the parent node of **{D, E}**.
- **Child Node:** The node which is the immediate successor of a node is called the child node of that node. Examples: **{D, E}** are the child nodes of **{B}.**
- **Root Node:** The topmost node of a tree or the node which does not have any parent node is called the root node. {**A}** is the root node of the tree. A non-empty tree must contain exactly one root node and exactly one path from the root to all other nodes of the tree.
- **Leaf Node or External Node:** The nodes which do not have any child nodes are called leaf nodes. **{I, J, K, F, G, H}** are the leaf nodes of the tree.
- **Ancestor of a Node:** Any predecessor nodes on the path of the root to that node are called Ancestors of that node. **{A,B}** are the ancestor nodes of the node **{E}**
- **Descendant:** A node x is a descendant of another node y if and only if y is an ancestor of x.
- **Sibling:** Children of the same parent node are called siblings. **{D,E}** are called siblings.
- **Level of a node:** The count of edges on the path from the root node to that node. The root node has level **0**.
- **Internal node:** A node with at least one child is called Internal Node.
- **Neighbour of a Node:** Parent or child nodes of that node are called neighbors of that node.
- **Subtree**: Any node of the tree along with its descendant.

Basic Terminologies In Tree Data Structure



Types of Tree data structures

- **Binary tree**: In a binary tree, each node can have a maximum of two children linked to it. Some common types of binary trees include full binary trees, complete binary trees, balanced binary trees, and degenerate or pathological binary trees.
- **Ternary Tree**: A Ternary Tree is a tree data structure in which each node has at most three child nodes, usually distinguished as “left”, “mid” and “right”.
- **N-ary Tree or Generic Tree**: Generic trees are a collection of nodes where each node is a data structure that consists of records and a list of references to its children. Unlike the linked list, each node stores the address of multiple nodes.



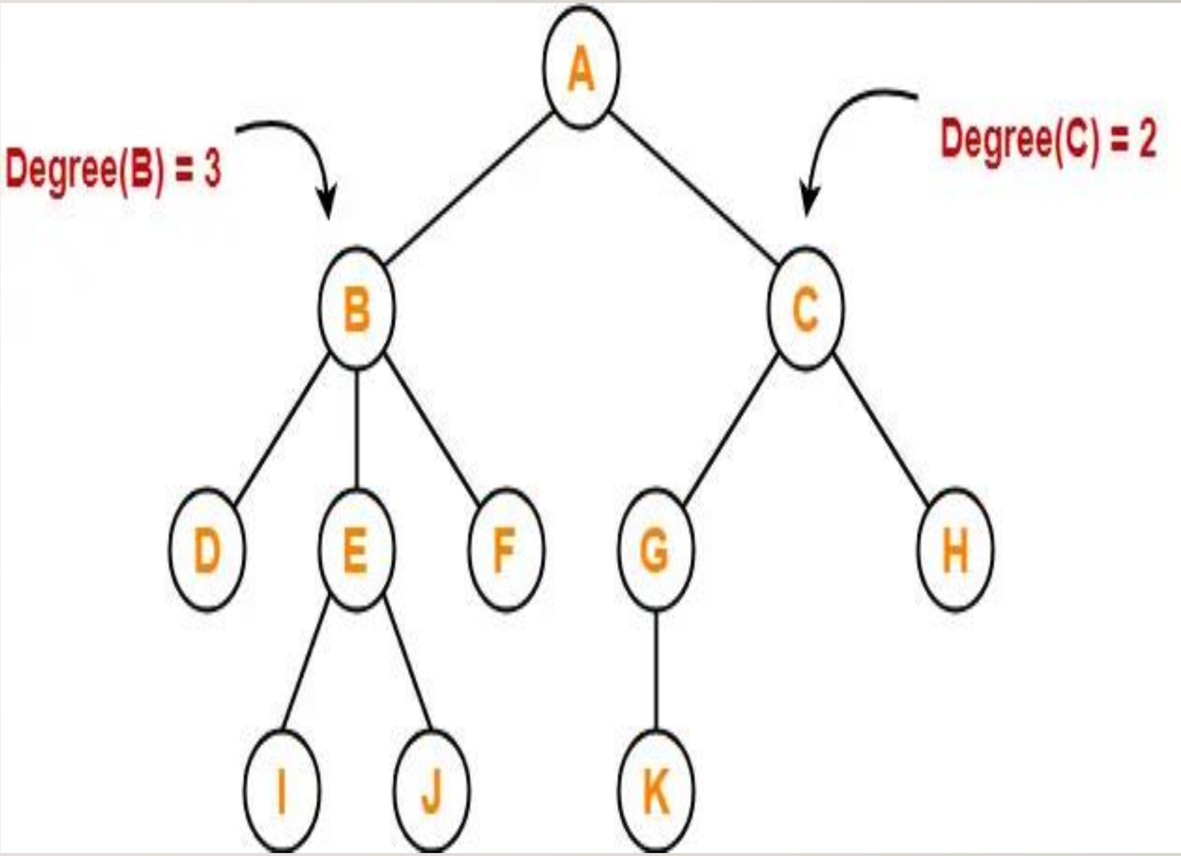
Basic Operations Of Tree Data Structure

- **Create** – create a tree in the data structure.
- **Insert** − Inserts data in a tree.
- **Search** − Searches specific data in a tree to check whether it is present or not.
- **Traversal**:
  - Depth-First-Search Traversal
  - Breadth-First-Search Traversal

Properties of Tree Data Structure

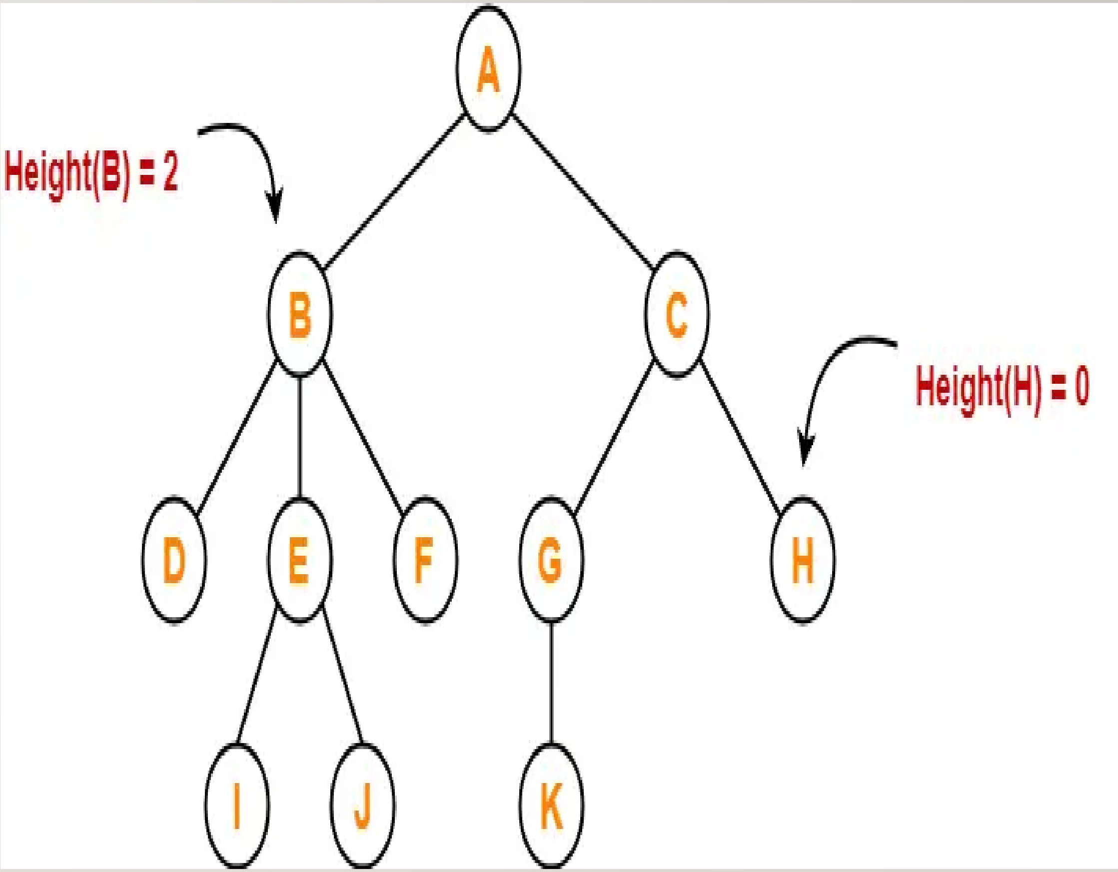
- **Number of edges:** An edge can be defined as the connection between two nodes. If a tree has N nodes then it will have (N-1) edges. There is only one path from each node to any other node of the tree.
- **Depth of a node:** The depth of a node is defined as the length of the path from the root to that node. Each edge adds 1 unit of length to the path. So, it can also be defined as the number of edges in the path from the root of the tree to the node.
- **Height of a node:** The height of a node can be defined as the length of the longest path from the node to a leaf node of the tree.
- **Height of the Tree:** The height of a tree is the length of the longest path from the root of the tree to a leaf node of the tree.
- **Degree of a Node:** The total count of subtrees attached to that node is called the degree of the node. The degree of a leaf node must be **0**. The degree of a tree is the maximum degree of a node among all the nodes in the tree.

- Degree of node A = 2
- Degree of node B = 3
- Degree of node C =
- Degree of node D =
- Degree of node E =
- Degree of node F =
- Degree of node G =
- Degree of node H =
- Degree of node I =
- Degree of node J =
- Degree of node K =



Height

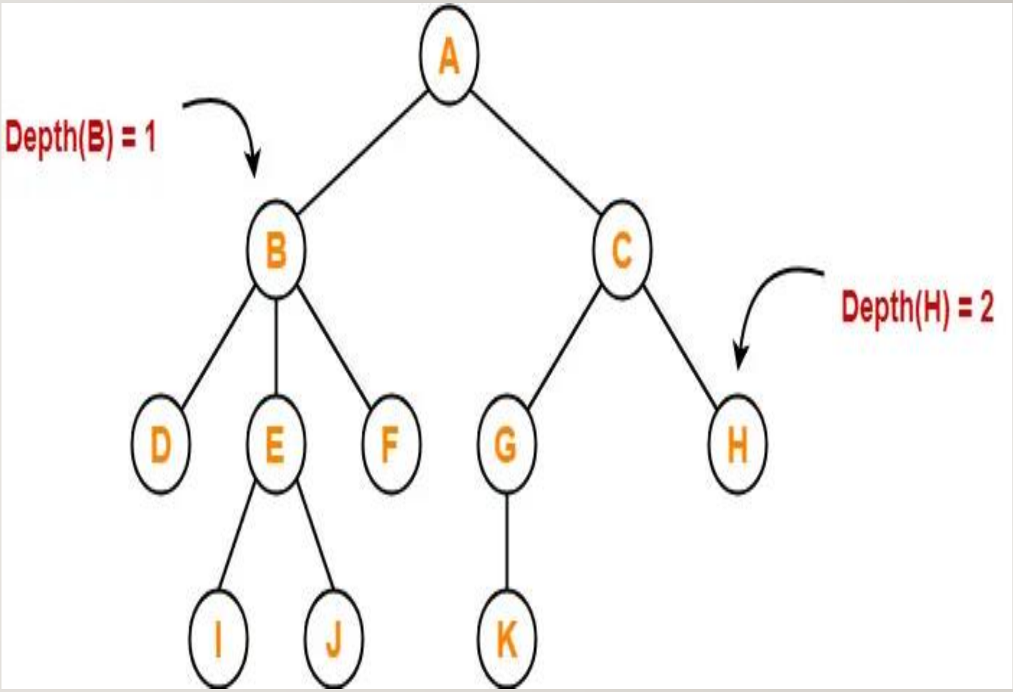
- Total number of edges that lies on the longest path from any leaf node to a particular node is called as **height of that node**.
- **Height of a tree** is the height of root node.
- Height of all leaf nodes = 0



Height of node A =
Height of node B = 2
Height of node C =
Height of node D =
Height of node E =
Height of node F =
Height of node G =
Height of node H = 0
Height of node I =
Height of node J =
Height of node K =

Depth

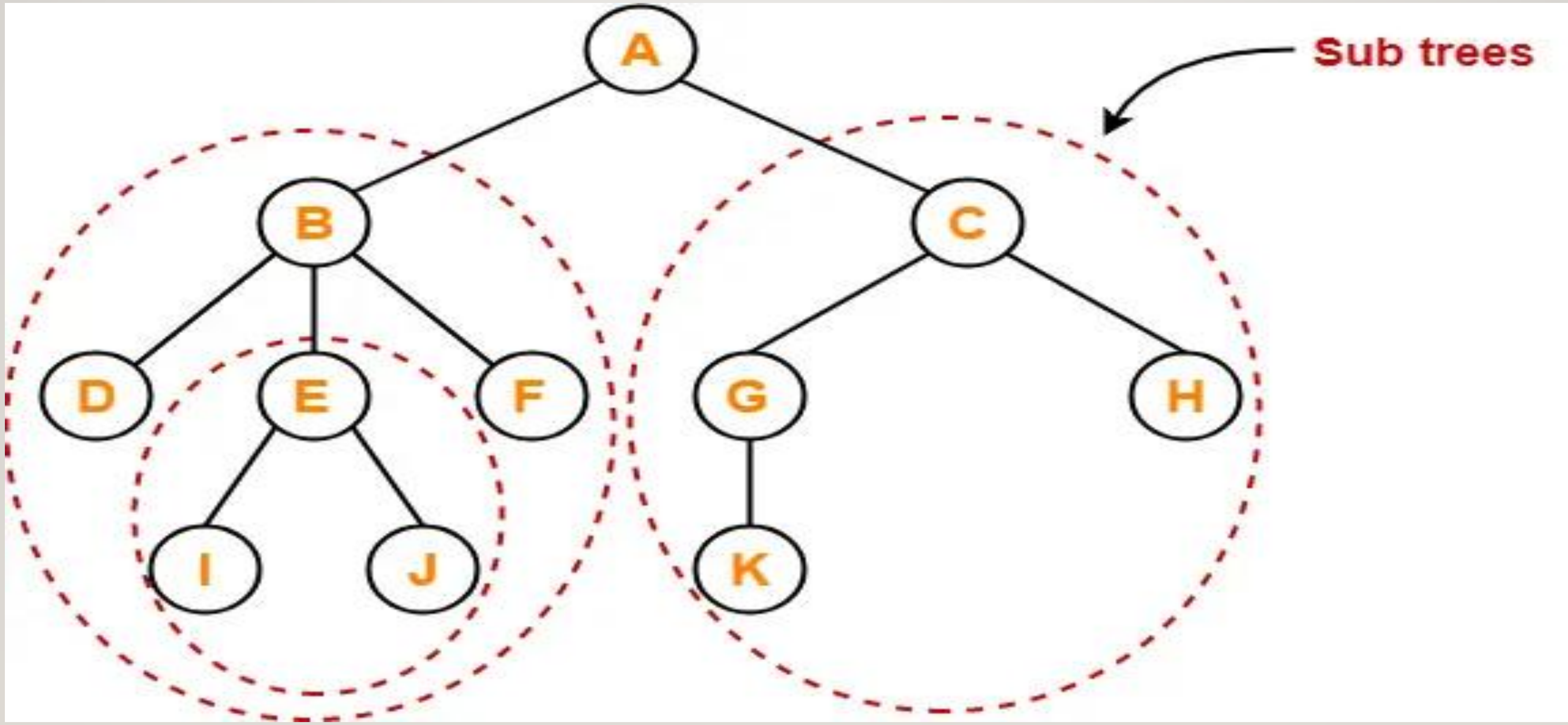
- Total number of edges from root node to a particular node is called as **depth of that node**.
- **Depth of a tree** is the total number of edges from root node to a leaf node in the longest path.
- Depth of the root node = 0
- The terms “level” and “depth” are used interchangeably.



- *Depth of node A =*
- *Depth of node B =*
- *Depth of node C =*
- *Depth of node D =*
- *Depth of node E =*
- *Depth of node F =*
- *Depth of node G =*
- *Depth of node H = 2*
- *Depth of node I =*
- *Depth of node J =*
- *Depth of node K =*

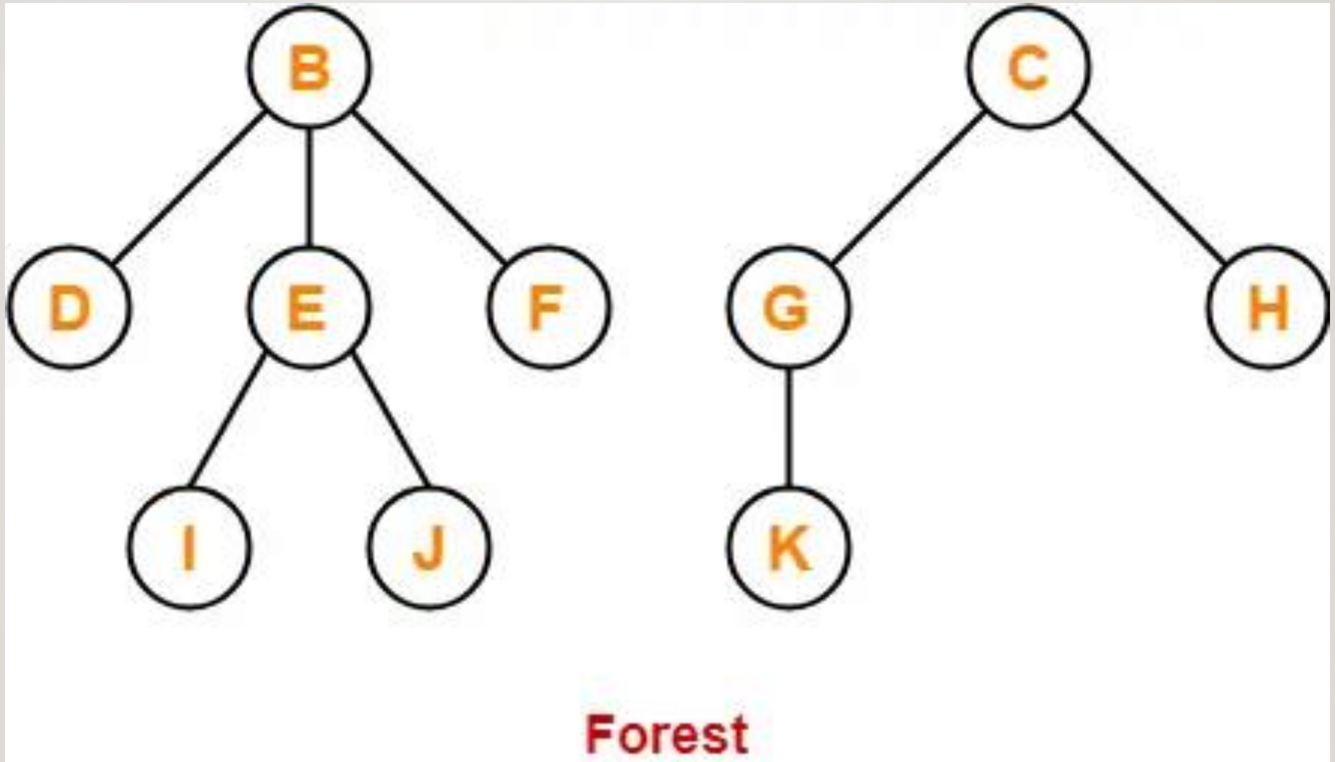
Subtree

- In a tree, each child from a node forms a subtree recursively.
- Every child node forms a subtree on its parent node.



Forest

A forest is a set of disjoint trees.

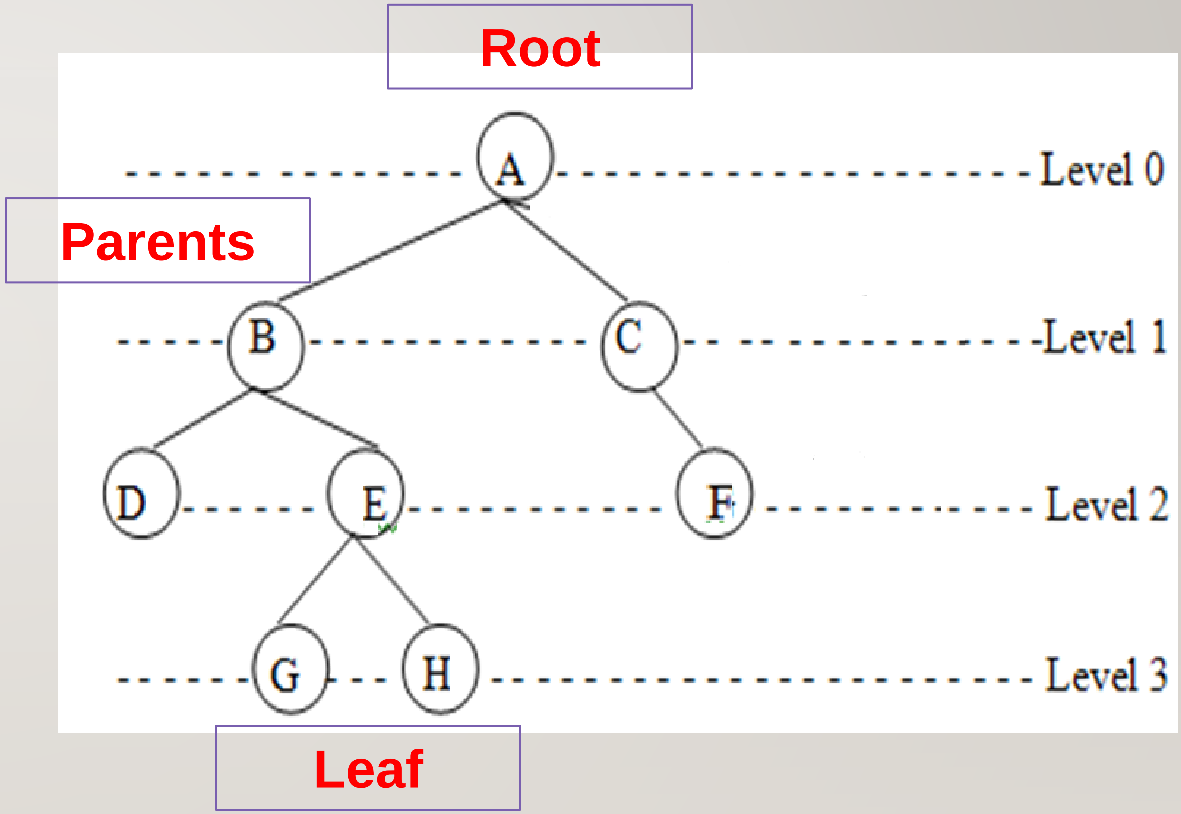


Binary Trees

Binary tree is a tree data structure(non-linear) in which each node can have at most two children which are referred to as the left child and the right child

Useful in modeling processes where

- a comparison or experiment has exactly two possible outcomes
- the test is performed repeatedly

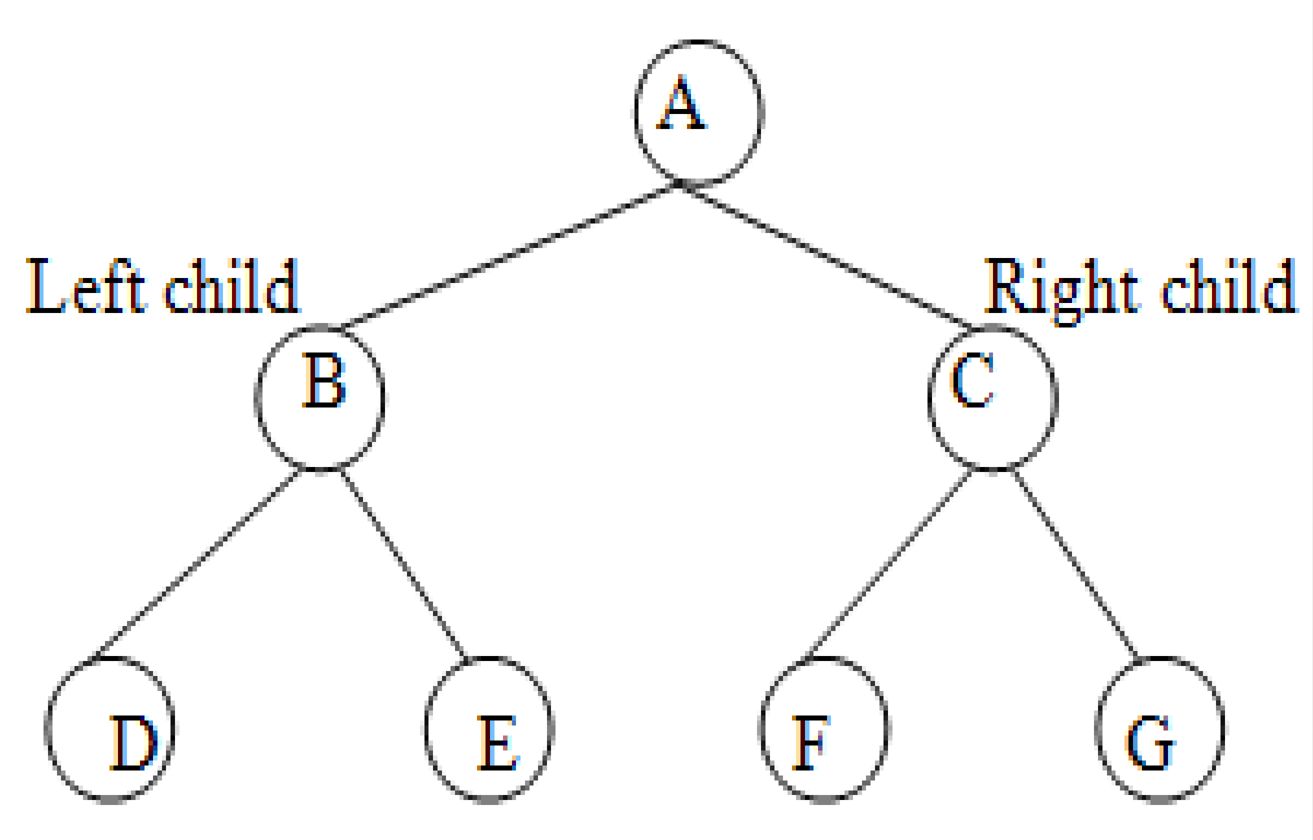


- Each node may have 0, 1, or 2 children.
- Node on the left as the ***left child*** and the node on the right as the ***right child***
- Binary tree is a set of nodes that either the tree is empty, or the tree is partitioned into three disjoint subsets: a single node R, the root; two possible empty sets that are binary trees, called left and right subtrees of R
- ***Degenerate tree*** occurs in which there is a single leaf node and each non-leaf node has only one child. It is equivalent to a linked list.
- A ***complete binary*** tree is a tree in which each level 0 to n-1 has full set of node and all leaf nodes at level n occupy the leftmost position in the tree.

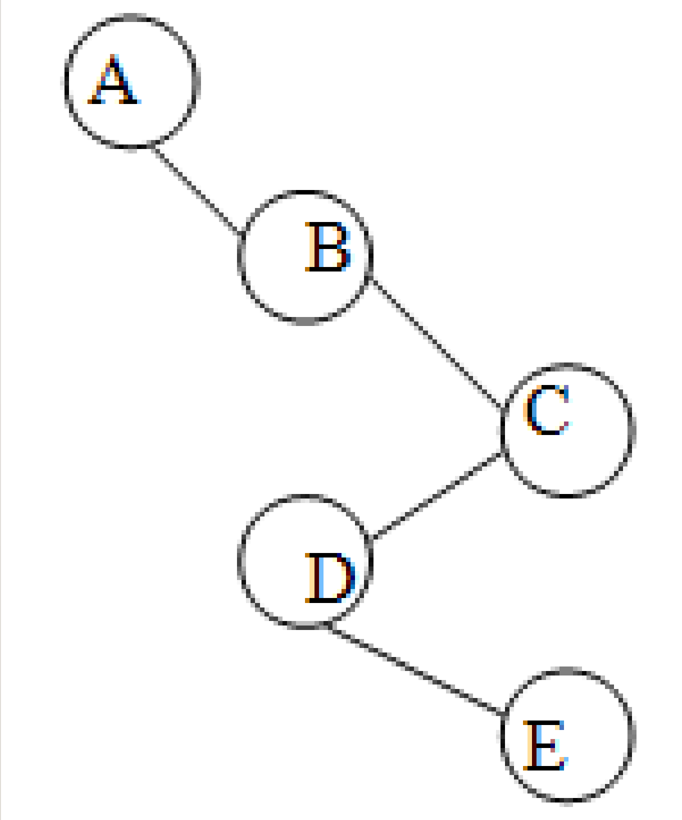
Properties of Binary Tree

- The maximum number of nodes at level **L** of a binary tree is $\mathbf{2^L}$
- The maximum number of nodes in a binary tree of height **H** is $\mathbf{2^H - 1}$
- Total number of leaf nodes in a binary tree = total number of nodes with 2 children + 1
- In a Binary Tree with **N** nodes, the minimum possible height or the minimum number of levels is $\mathbf{Log_2(N+1)}$
- A Binary Tree with **L** leaves has at least **| Log2L |+ 1** levels

Complete Binary tree



Degenerate tree



ARRAY IMPLEMENTATION OF BINARY TREE

Assume you have the below binary tree: