كلية علوم الحاسوب والرياضيات
College of Computer Science & Mathematics


# Software Engineering Dept.- Second year

*Tree Data Structure*

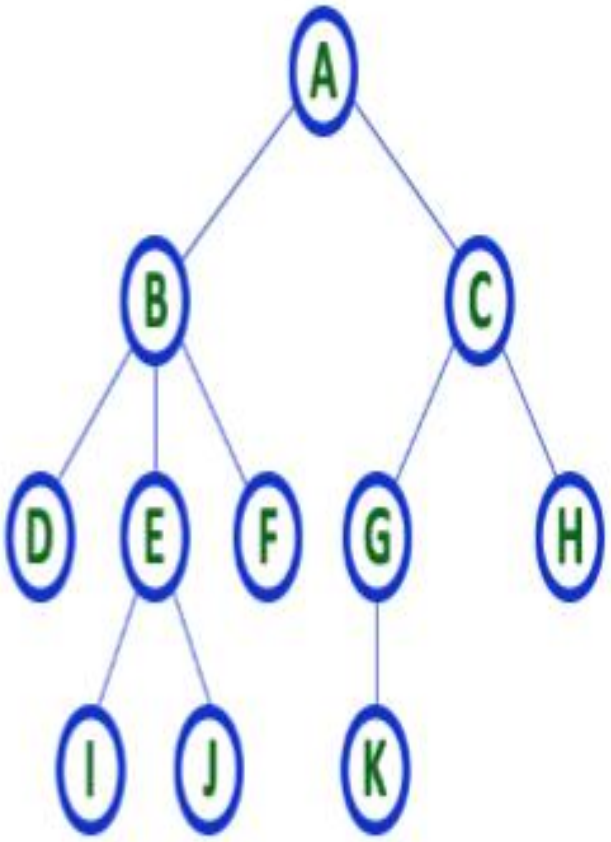
*by*

Dr.laheeb M. Ibrahim

# Tree data structure

- **Tree data structure** is a collection of data (Node) which is organized in hierarchical structure.
- In tree data structure, every individual element is called as **Node**.
- Node in a tree data structure, stores the actual data of that particular element and link to next element in hierarchical structure.
- In a tree data structure, if we have N number of nodes then we can have a maximum of N-1 number of links.

# Example for tree data structure




TREE with 11 nodes and 10 edges

- In any tree with 'N' nodes there will be maximum of 'N-1' edges

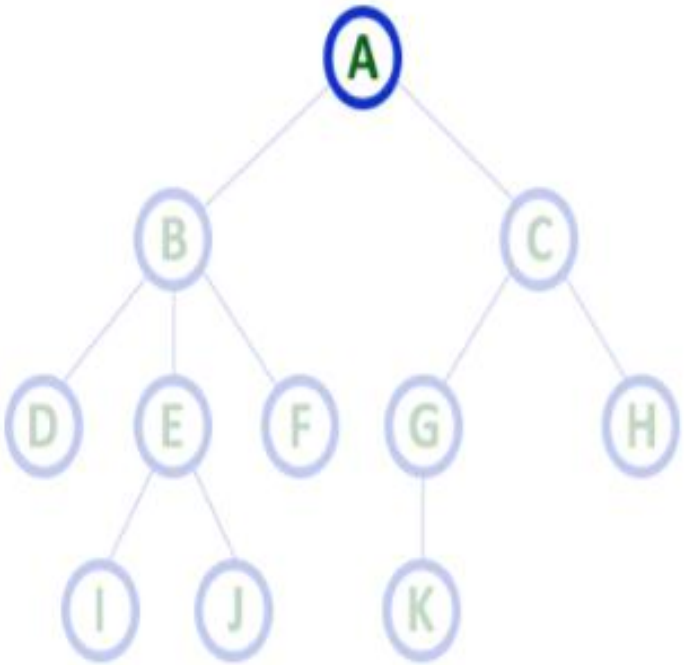
- In a tree every individual element is called as 'NODE'

## Tree Terminology

In a tree data structure, we use the following terminology.

1. **Root:** In a tree data structure, the first node is called as Root Node. Every tree **must have root node**. In any tree, there **must be only one root** node.

# Example

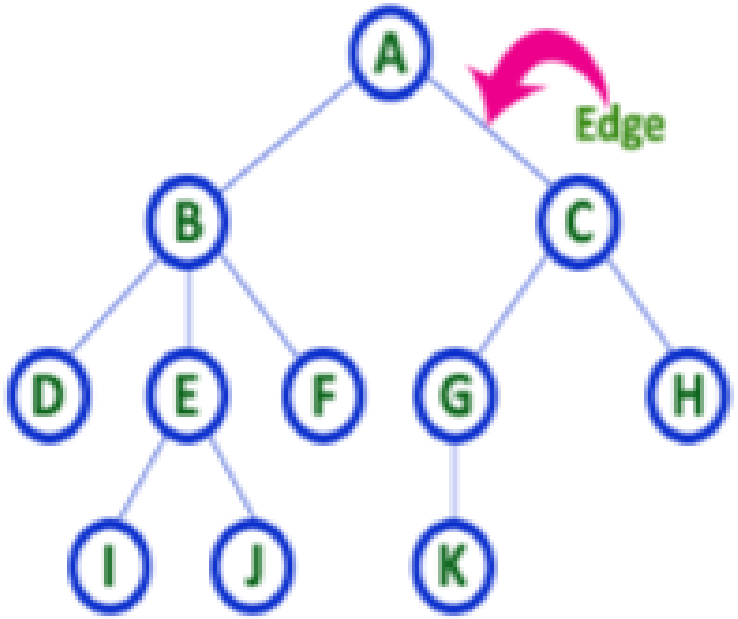



Here ‘A’ is the ‘root’ node

- In any tree the first node is called as ROOT node

# 2. Edge

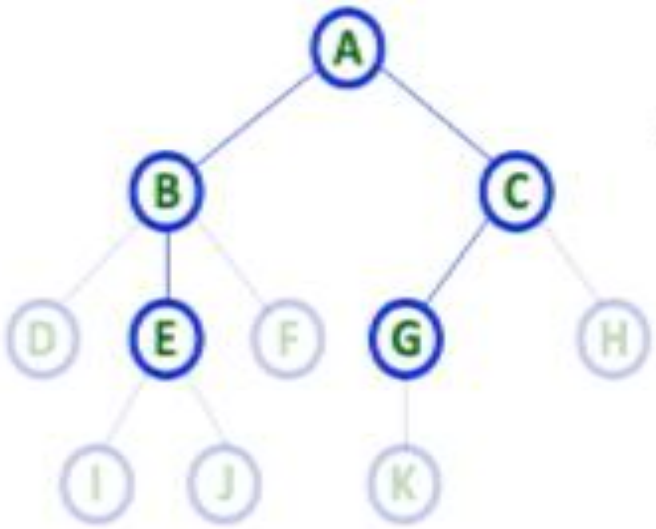
In a tree data structure, the connecting link between any two nodes is called as **EDGE**. In a tree with 'N' number of nodes there will be a maximum of '**N-1**' number of edges.




- In any tree, 'Edge' is a connecting link between two nodes.

# 3. Parent

In a tree data structure, the node which is predecessor of any node is called as **PARENT NODE**. In simple words, the node which has branch from it to any other node is called as parent node. Parent node can also be defined as "**The node which has child / children**".

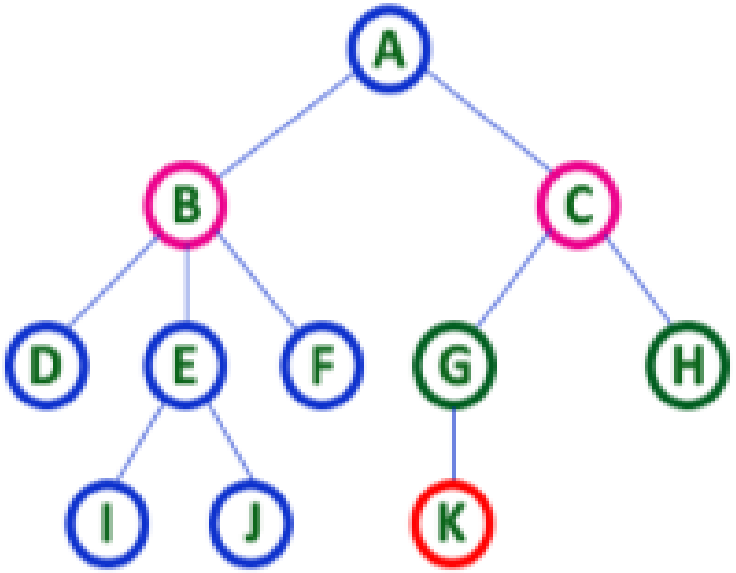



**Here A, B, C, E & G are Parent nodes**

- In any tree the node which has child / children is called '**Parent**'
- A node which is predecessor of any other node is called '**Parent**'

# 4. Child

In a tree data structure, the node which is descendant of any node is called as **CHILD Node**. In simple words, the node which has a link from its parent node is called as child node. In a tree, any parent node can have any number of child nodes. In a tree, all the nodes except root are child nodes.




Here B & C are Children of A

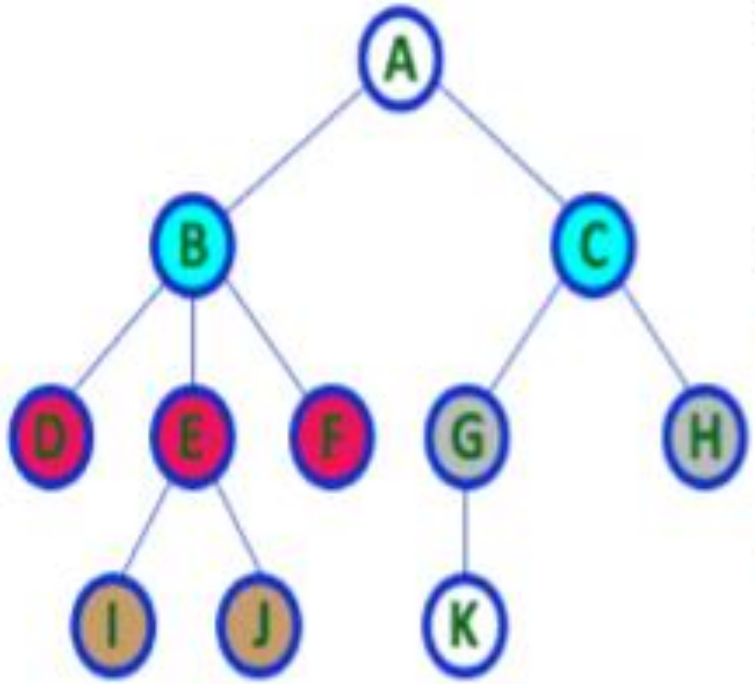
Here G & H are Children of C

Here K is Child of G

- descendant of any node is called as CHILD Node

# 5. Siblings

In a tree data structure, nodes which belong to same Parent are called as **SIBLINGS**.
In simple words, the nodes with same parent are called as Sibling nodes.

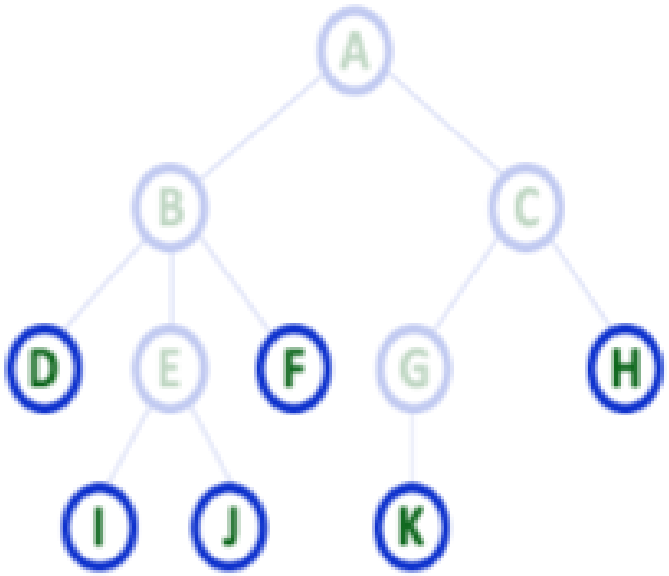



Here B & C are Siblings
Here D E & F are Siblings
Here G & H are Siblings
Here I & J are Siblings

- In any tree the nodes which has same Parent are called 'Siblings'
- The children of a Parent are called 'Siblings'

# 6. Leaf

In a tree data structure, the node which does not have a child is called as **LEAF Node**. In simple words, a leaf is a node with no child. In a tree data structure, the leaf nodes are also called as **External Nodes**. External node is also a node with no child. In a tree, leaf node is also called as '**Terminal**' node.

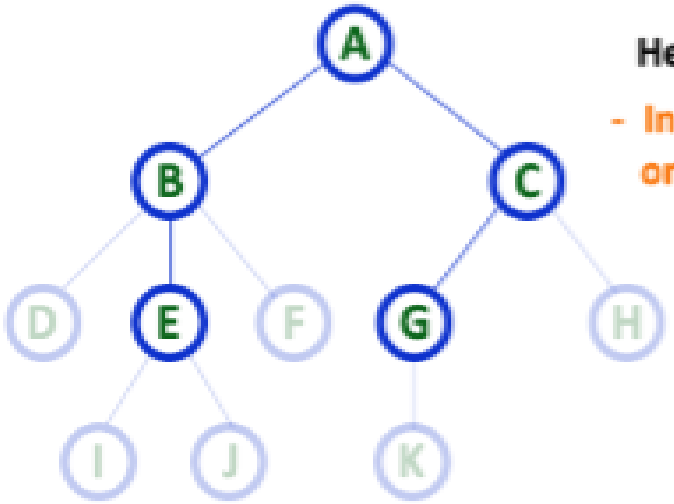



**Here D, I, J, F, K & H are Leaf nodes**

- In any tree the node which does not have children is called 'Leaf'
- A node without successors is called a 'leaf' node

# 7. Internal Nodes

In a tree data structure, the node which has at least one child is called as **INTERNAL Node**. In simple words, an internal node is a node with at least one child. In a tree data structure, nodes other than leaf nodes are called as **Internal Nodes**. **The root node is also said to be Internal Node** if the tree has more than one node. Internal nodes are also called as '**Non-Terminal**' nodes.



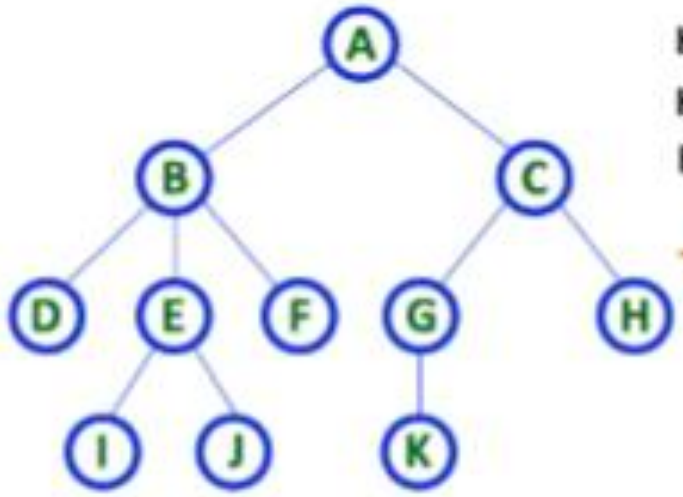

Here A, B, C, E & G are Internal nodes

- In any tree the node which has atleast one child is called 'Internal' node

- Every non-leaf node is called as 'Internal' node

# 8. Degree

In a tree data structure, the total number of children of a node is called as **DEGREE** of that Node. In simple words, the Degree of a node is total number of children it has. The highest degree of a node among all the nodes in a tree is called as '**Degree of Tree**'




Here Degree of B is 3

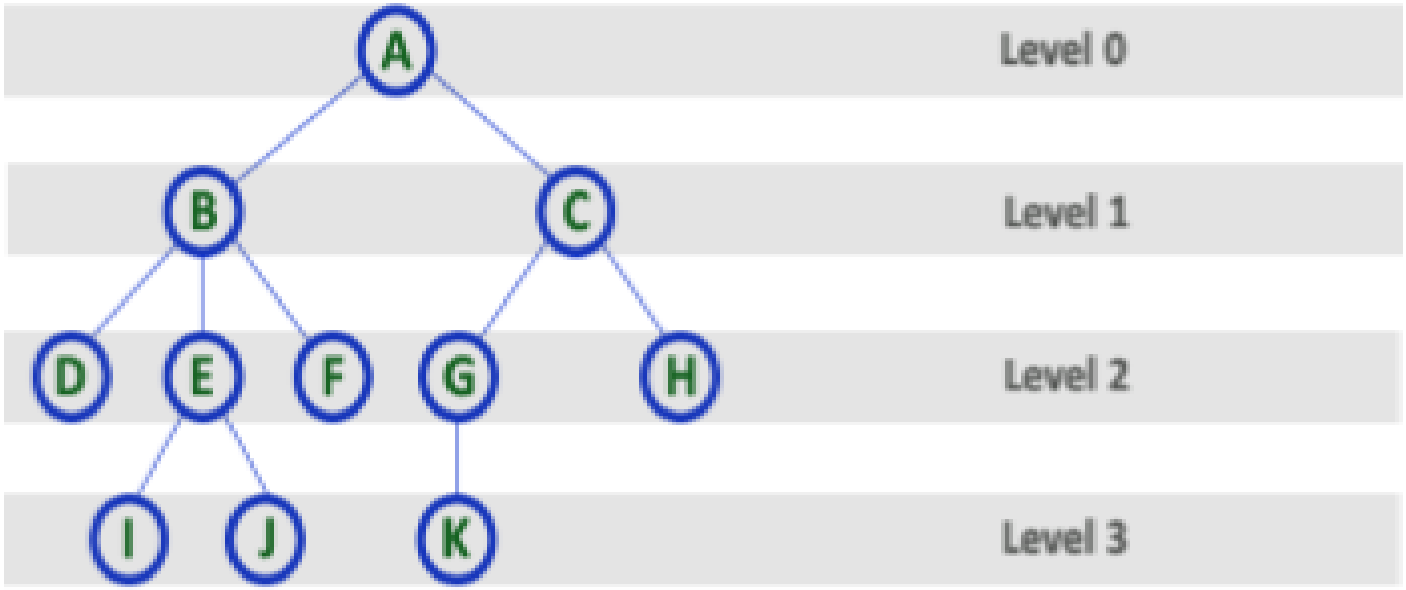
Here Degree of A is 2

Here Degree of F is 0

- In any tree, 'Degree' a node is total number of children it has.

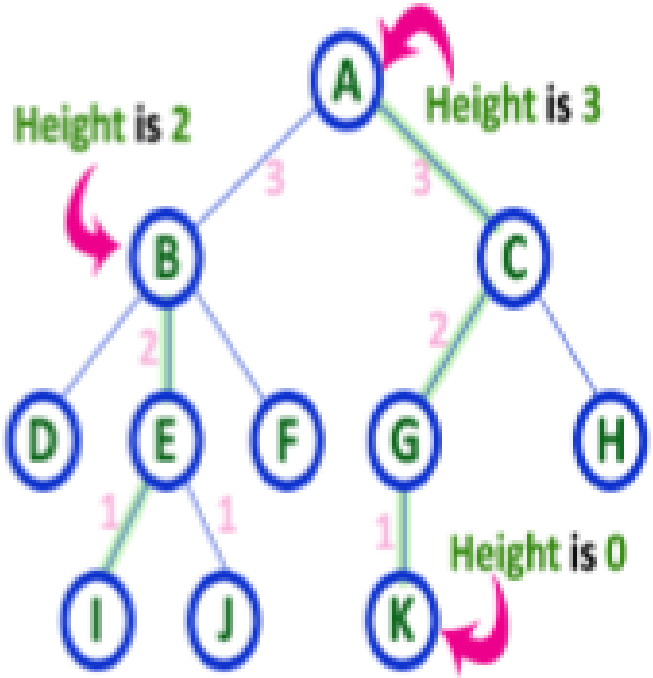
# 9. Level

In a tree data structure, the root node is said to be at Level 0 and the children of root node are at Level 1 and the children of the nodes which are at Level 1 will be at Level 2 and so on... In simple words, in a tree each step from top to bottom is called as a Level and the Level count starts with '0' and incremented by one at each level (Step).



# 10. Height

In a tree data structure, the total number of egdes from leaf node to a particular node in the longest path is called as **HEIGHT** of that Node. In a tree, height of the root node is said to be **height of the tree**. In a tree, **height of all leaf nodes is '0'.**



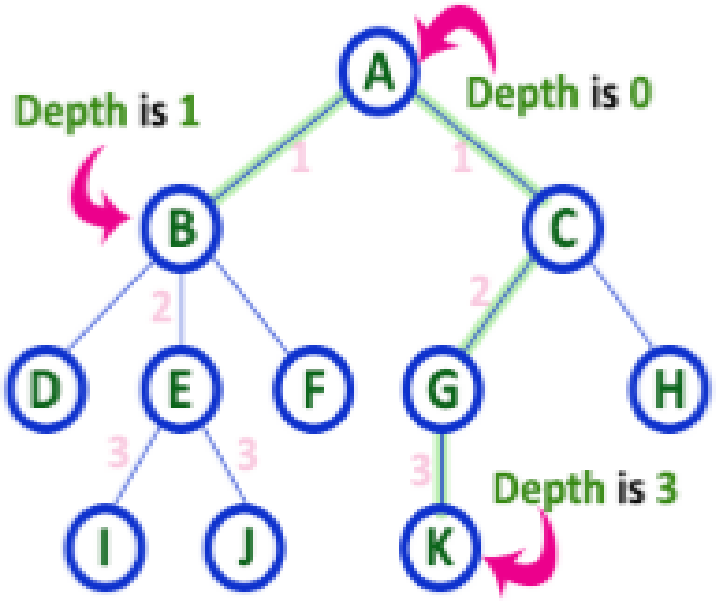

Here Height of tree is 3

- In any tree, 'Height of Node' is total number of Edges from leaf to that node in longest path.

- In any tree, 'Height of Tree' is the height of the root node.

# 11. Depth

In a tree data structure, the total number of egdes from root node to a particular node is called as **DEPTH** of that Node. In a tree, the total number of edges from root node to a leaf node in the longest path is said to be **Depth of the tree**. In simple words, the highest depth of any leaf node in a tree is said to be depth of that tree. In a tree, **depth of the root node is '0'.**

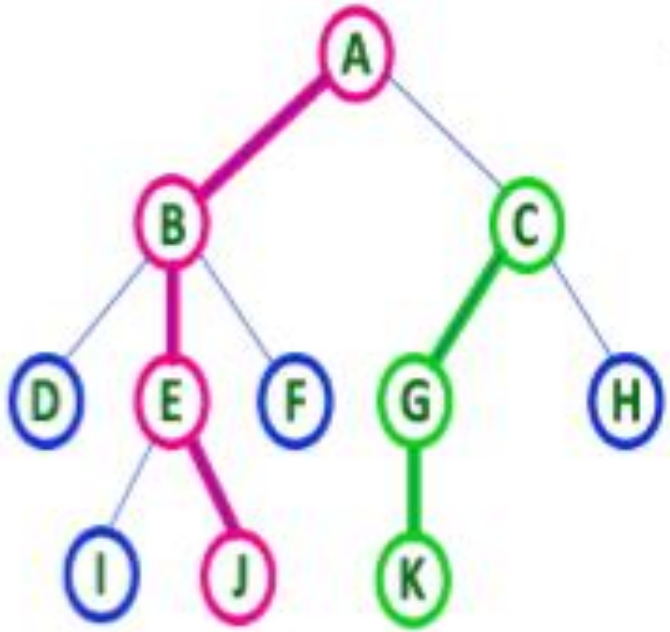



Here Depth of tree is 3

- In any tree, 'Depth of Node' is total number of Edges from root to that node.
- In any tree, 'Depth of Tree' is total number of edges from root to leaf in the longest path.

# 12. Path

In a tree data structure, the sequence of Nodes and Edges from one node to another node is called as **PATH** between that two Nodes. **Length of a Path** is total number of nodes in that path. In below example **the path A - B - E - J has length 4**.




- In any tree, 'Path' is a sequence of nodes and edges between two nodes.

Here, 'Path' between A & J is

A - B - E - J

Here, 'Path' between C & K is

C - G - K

# 13. Sub Tree

In a tree data structure, each child from a node forms a subtree recursively. Every child node will form a subtree on its parent node.