جامـــــعة المـــــستـقبـل
AL MUSTAQBAL UNIVERSITY

كلية العلـوم

قسـم علوم الذكاء الاصطناعي

المحاضرة الثالثة

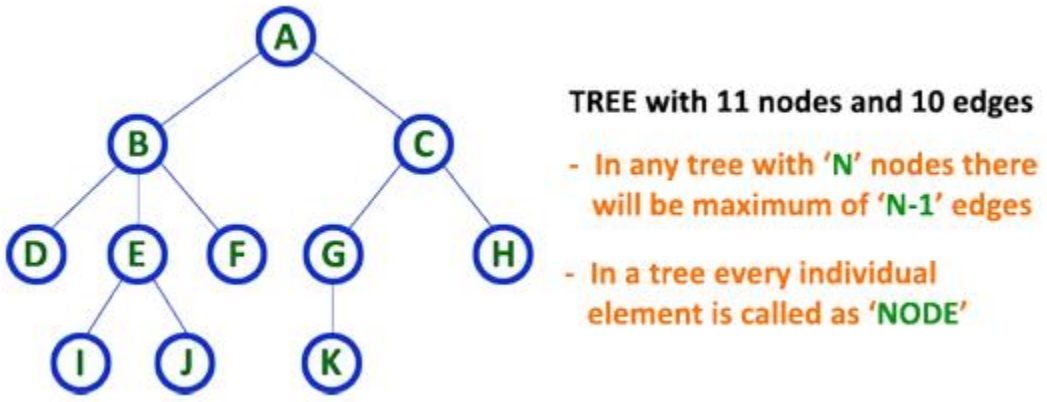
المادة: Searching and Sorting Algorithms
المرحلة: الثانية
اسم الاستاذ: م.م اية محمد حسين محمد علي

Trees

A **Tree** is a hierarchical data structure made of nodes connected by edges.
It is widely used in searching and sorting algorithms because it allows fast data access and organization.

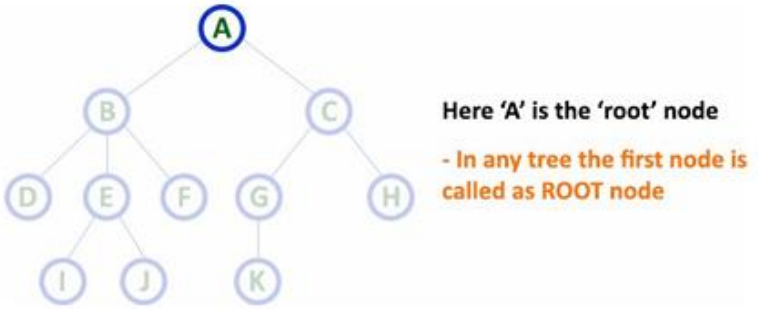
In many search and sort methods, special types of trees are used to improve efficiency.

Example:

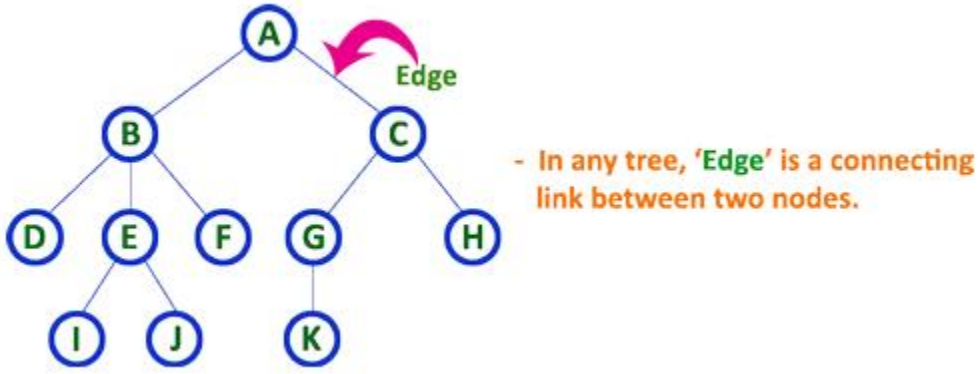


In a tree data structure, we use the following terminology...

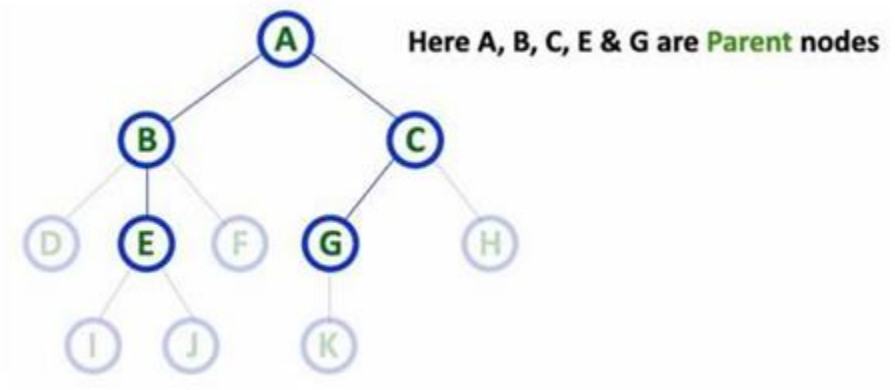
1. Root : In a tree data structure, the first node is called as Root Node. Every tree must have root node. We can say that root node is the origin of tree data structure. In any tree, there must be only one root node.



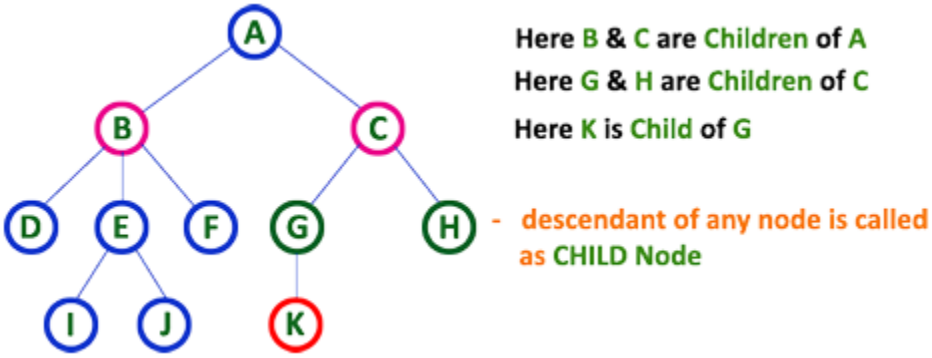
2. Edge: In a tree data structure, the connecting link between any two nodes is called as EDGE. In a tree with 'N' number of nodes there will be a maximum of 'N-1' number of edges.



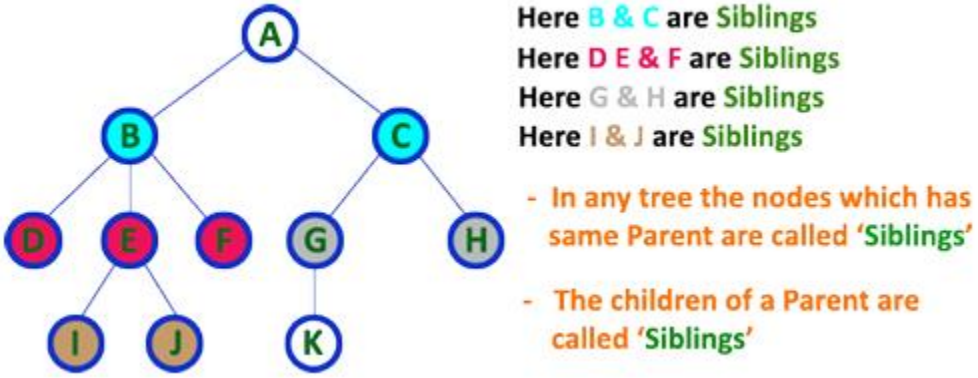
3. Parent: In a tree data structure, the node which is predecessor of any node is called as PARENT NODE. In simple words, the node which has branch from it to any other node is called as parent node. Parent node can also be defined as "The node which has child / children".



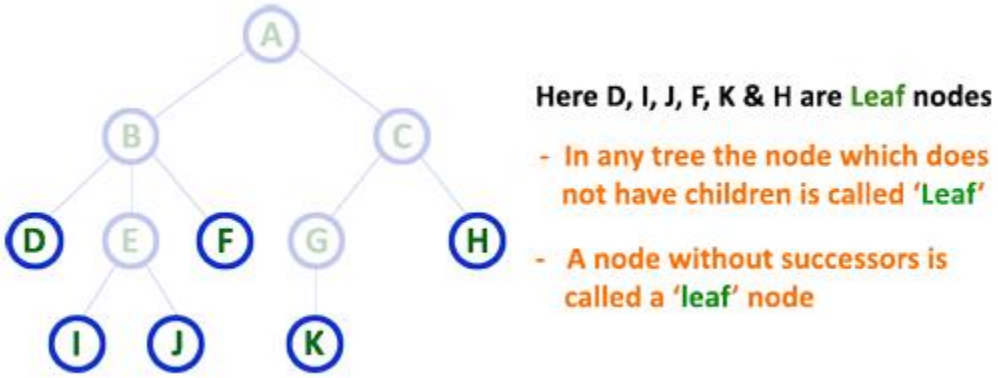
4. Child: In a tree data structure, the node which is descendant of any node is called as CHILD Node. In simple words, the node which has a link from its parent node is called as child node. In a tree, any parent node can have any number of child nodes. In a tree, all the nodes except root are child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



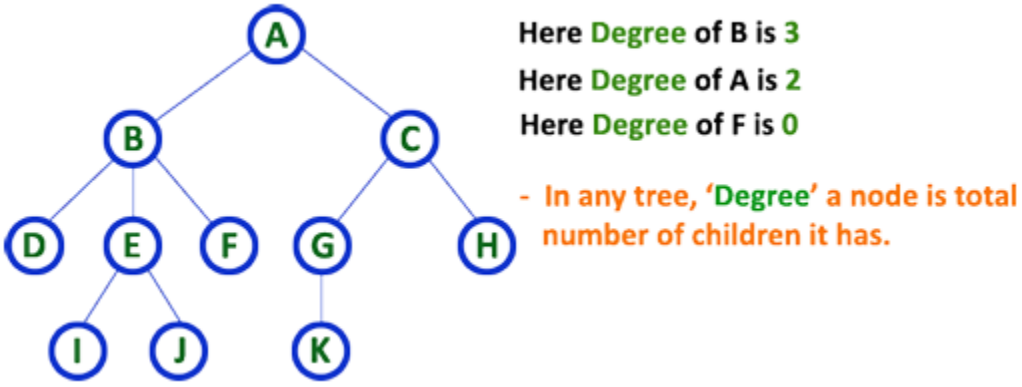
5. Siblings: In a tree data structure, nodes which belong to same Parent are called as SIBLINGS. In simple words, the nodes with same parent are called as Sibling nodes.



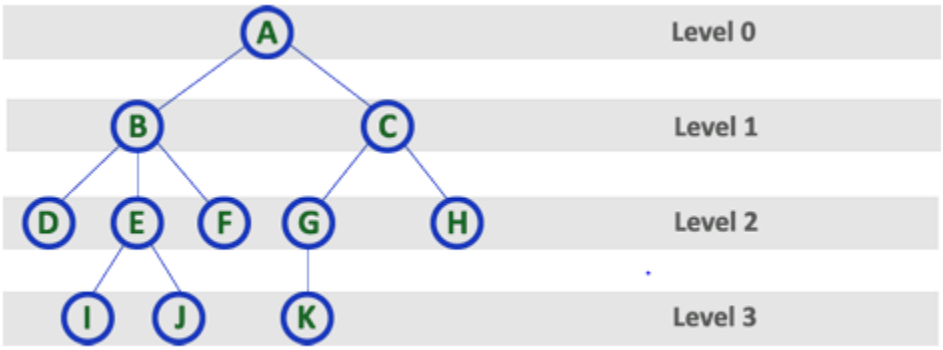
6. Leaf: In a tree data structure, the node which does not have a child is called as LEAF Node. In simple words, a leaf is a node with no child. leaf node is also called as 'Terminal' node.



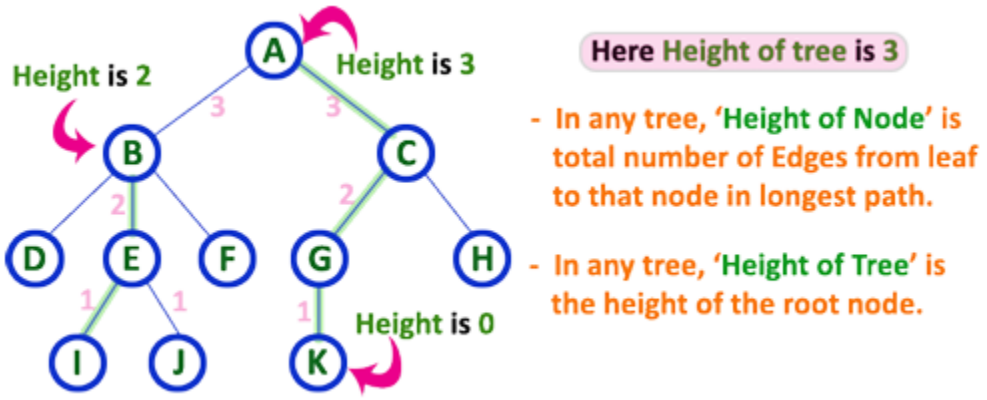
7. Degree: In a tree data structure, the total number of children of a node is called as DEGREE of that Node. In simple words, the Degree of a node is total number of children it has. The highest degree of a node among all the nodes in a tree is called as 'Degree of Tree'



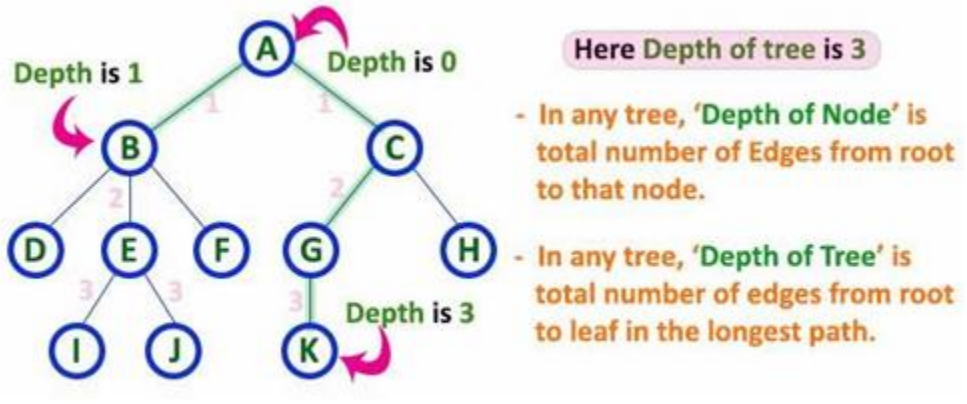
8. Level: In a tree data structure, the root node is said to be at Level 0 and the children of root node are at Level 1 and the children of the nodes which are at Level 1 will be at Level 2 and so on... In simple words, in a tree each step from top to bottom is called as a Level and the Level count starts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



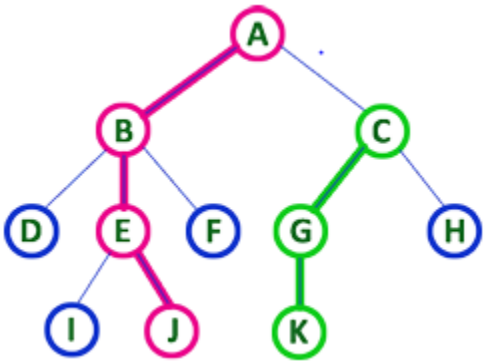
9. Height: In a tree data structure, the total number of edges from leaf node to a particular node in the longest path is called as HEIGHT of that Node. In a tree, height of the root node is said to be height of the tree. In a tree, height of all leaf nodes is '0'.



10. Depth: In a tree data structure, the total number of edges 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'.



11. 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 length4.

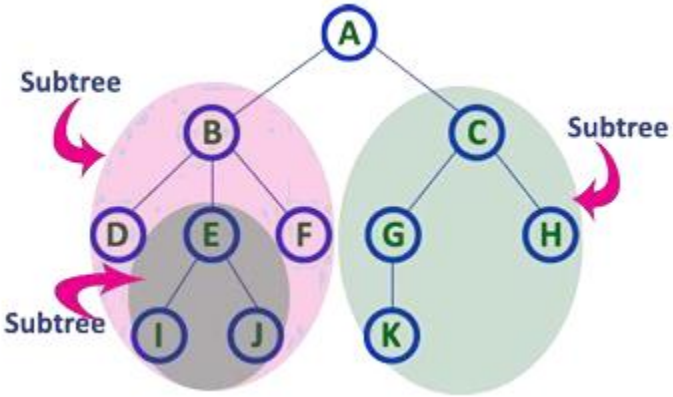


- In any tree, 'Path' is a sequence of nodes and edges between two nodes.

Here, 'Path' between A & J is
A - B - E - J

Here, 'Path' between C & K is
C - G - K

12. 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.



Why Trees Are Useful?

Faster searching.
Faster sorting.
Better data organization.
Reduce number of comparisons..
Used in databases and indexing
Used in priority queues.