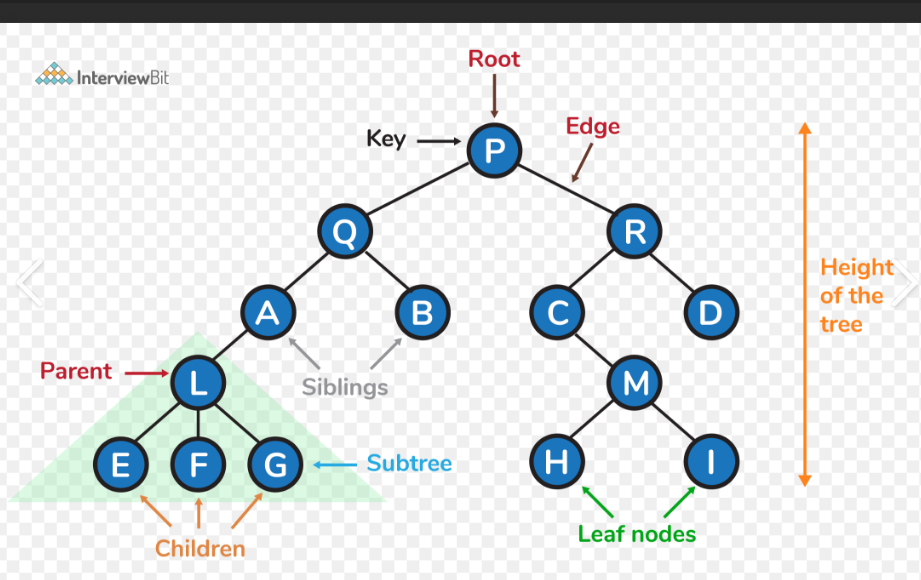
Linear and Non-linear data Structure

1. Linear Data Structure :

* 1. **linear data structures arrange data in a sequential manner** .
  2. **A linear data structure is a data structure that has data elements in sequential order. In a linear data structure, the adjacent elements are attached to each other. However, these data structures do not make better utilization of memory. Therefore, it can lead to memory**
  3. **wastage.**

1. Non-Linear Data Structure :
   1. **nonlinear data structures arrange data in a hierarchical manner, creating a relationship among the data elements.**
   2. **Non Linear data structure stores data in a non-sequential manner. It forms a hierarchical relationship among the child elements and parent elements. In other words, the data items are attached to each other creating a relationship between them. It is not possible to insert elements, delete elements or go through the elements in sequential order. Usually, these data structures are more memory efficient.**

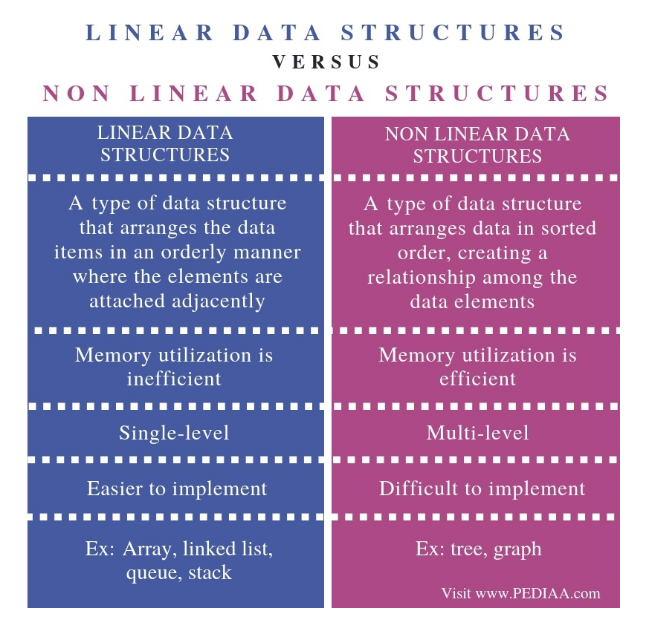
**Tree Data Structure :**

****

1. Non-linear data structure
2. Represents relationship between nodes
3. Collections of entities called nodes
4. Nodes are connected by edges

Teminologies :

1. Node : The individual elements of tree is called node .
2. Root : Top elements of tree .
3. Edge/Link : Connection between two nodes .
4. Parent Node : The node which has a brach/edge from it to any other node is called parent node .
5. Siblings : Nodes which belongs two same parents are called as siblings.
6. Leaf node / external node / terminal nodes : Node which does not have a child .
7. Internal node : Node which have atleast one child.
8. Path : The sequence of node and edges from one node to another is called path between two nodes .

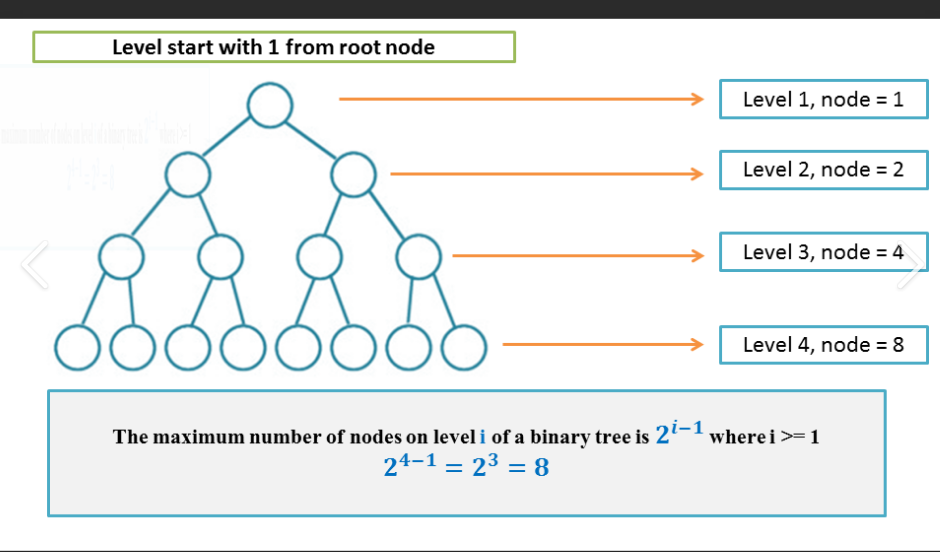


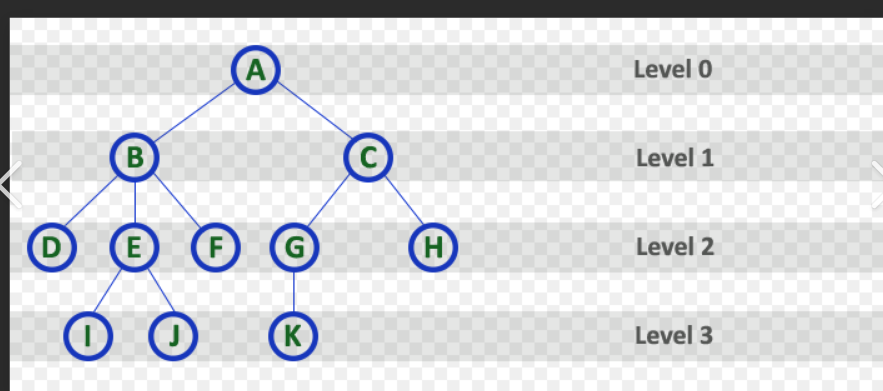
Characteristics of Tree data structure :

1. In a tree if we have N nodes then we have N-1 edges / links .
2. Tree is a recursive data structure .

Tree Terminologies :

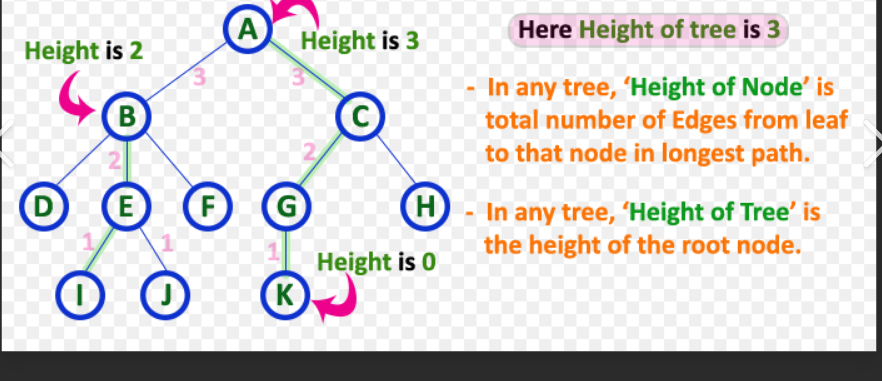
1. Degree of a node : Total number of childrens of that node .
2. Degree of Tree : Highest degree of a node among all the nodes in the tree .
3. Level of tree :





1. Hight
   1. Height of node :

*The* ***height*** *of a node is the number of edges present in the longest path connecting that node to a leaf node.*

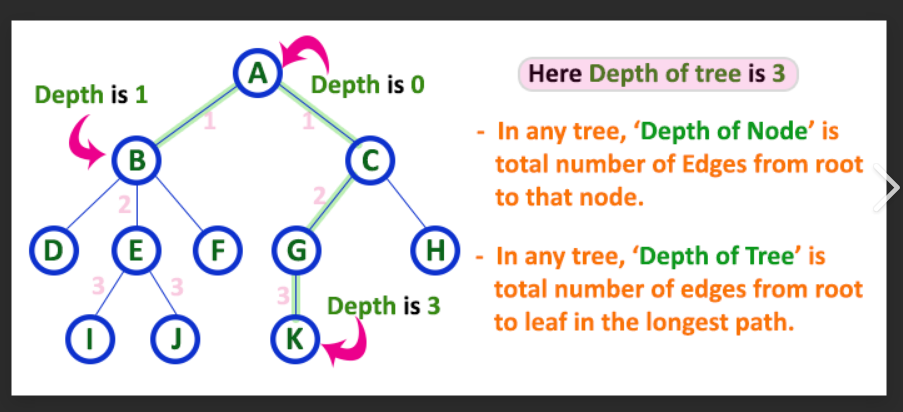


1. Depth of Node :

*sskkkaa*

*The* ***depth*** *of a node is the number of edges present in path from the root node of a* ***tree*** *to that node.*

1. Depth of tree :



Applications :

1. Searching
2. Heap Sort etc.

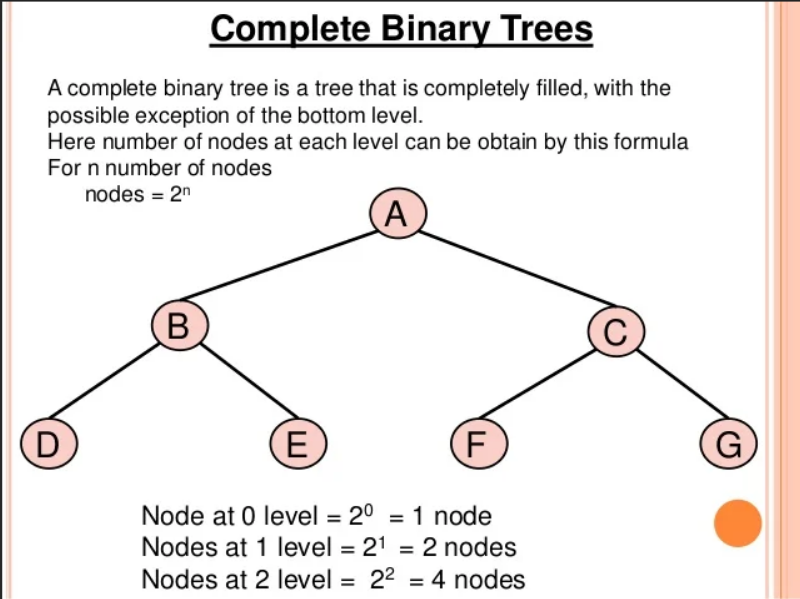
Types of Tree :

1. General Tree :

A General Tree is one of the basic forms of [Tree Data Structure](https://www.javatpoint.com/tree). In General Tree, each node can have either zero or more than zero nodes associated with it as the child nodes.

1. Binary Tree :

A [binary tree](https://www.javatpoint.com/binary-tree) can be defined as one of the trees in which only maximum two children can be added to each parent node.

* 1. Full Binary Tree
     1. Type of binary tree in which every node has 0 or 2 child nodes (not 1 node ).
  2. Complete Binary Tree .
     1. A complete binary tree is a special type of binary tree where all the levels of the tree are filled completely except the lowest level nodes which are filled from as left as possible.
     2. A complete binary tree is a special type of binary tree where all the levels of the tree are filled completely except the lowest level nodes which are filled from as left as possible.
  3. Perfect Binary Tree
     1. Type of binary tree in which all internal node have contain two children and all leaf nodes are present in same level .
  4. Balenced Binary Tree
     1. A balanced binary tree, also referred to as a {height-balanced} binary tree, is defined as a binary tree in which the height of the left and right subtree of any node differ by not more than 1.
     2. Formula : [Height of Left SubTree - Height of Right SubTree]
  5. Pathalogical Binary Tree / Degenerate Binary Tree :
     1. Degenerate Binary Trees If in a binary tree **each node contains only one child node either on the left side or the right side of the tree**, it is known as a degenerate binary tree. Degenerate binary trees are equal to linked lists in terms of performance.