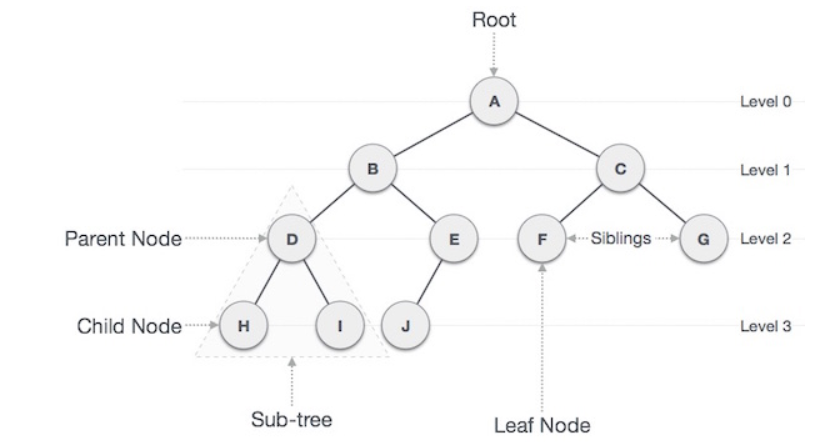
**TREES CHEAT SHEET:-**

Parts of a tree:-



**IMPORTANT TREE DEFINITIONS**

* **Path** − Path refers to the sequence of nodes along the edges of a tree.
* **Root** − The node at the top of the tree is called root. There is only one root per tree and one path from the root node to any node.
* **Parent** − Any node except the root node has one edge upward to a node called parent.
* **Child** − The node below a given node connected by its edge downward is called its child node.
* **Leaf** − The node which does not have any child node is called the leaf node.
* **Subtree** − Subtree represents the descendants of a node.
* **Visiting** − Visiting refers to checking the value of a node when control is on the node.
* **Traversing** − Traversing means passing through nodes in a specific order.
* **Levels** − Level of a node represents the generation of a node. If the root node is at level 0, then its next child node is at level 1, its grandchild is at level 2, and so on.
* **keys** − Key represents a value of a node based on which a search operation is to be carried out for a node
* **Height** - It can be defined as the longest path downwards between the root and the leaf.
* **Degree of a tree -** Maximum number of child node to a root or parent is called as degree of the tree
* **Binary tree -** A tree with a maximum degree of 2 is called as binary tree.
* **N-ary tree -** A tree with a maximum degree of N is called N-ary tree.

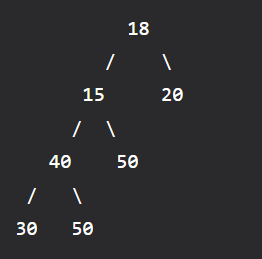
**IMPORTANT FORMULA**

* **Total no of edges=**Total no of nodes -1
* **Total leaf nodes =**(Total no of nodes having 2 children) +1
* **Max no of element at level L = 2^(L)** if L starts from 0 or **2^(L-1)** if L starts from 1
* **Max no of nodes for tree with level l=2^(L+1)-1** if L starts from 0 or **2^(L)-1** if power starts from 1
* **Height of tree is H ,then max node=2^(H+1)-1** if H starts from 0 or **2^(H)-1** if H starts from 1
* **Minimum number of nodes in a Binary tree of height h would be =(H+1)** if H starts from 0 or **(H)** if height starts from 1
* **Binary tree with N nodes min possible height =(Log2(N+1)-1)** if h starts from 0 or **(Log2(N+1))** if h starts from 1

**TYPES OF BINARY TREES:-**

**Full Binary Tree:**

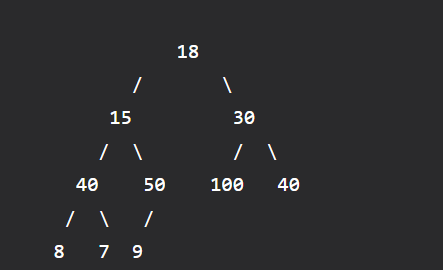
* A Binary Tree is a full binary tree if every node has 0 or 2 children
* The degree must be either 0 or 2



**Complete Binary Tree:-**

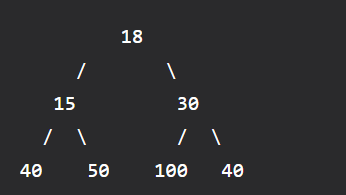
1. A Binary Tree is a Complete Binary Tree if all the levels are completely filled except possibly the last level and the last level has all keys as left as possible.
2. A complete binary tree is just like a full binary tree, but with two major differences:

* Every level must be completely filled
* All the leaf elements must lean towards the left.
* The last leaf element might not have a right sibling i.e. a complete binary tree doesn’t have to be a full binary tree.

****

**Perfect Binary Tree:-**

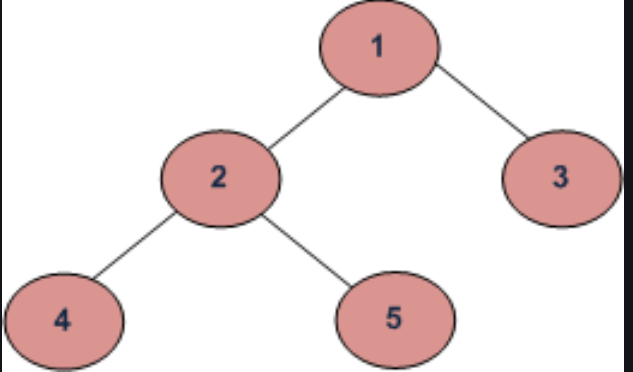
A Binary tree is a Perfect Binary Tree in which all the internal nodes have two children and all leaf nodes are at the same level.



**Traversal techniques**

**There are three types of traversal:**

1. **Postorder**
2. **Preorder**
3. **Inorder**

****

**Depth First Traversals:**

**(a) Inorder (Left, Root, Right) : 4 2 5 1 3**

**(b) Preorder (Root, Left, Right) : 1 2 4 5 3**

**(c) Postorder (Left, Right, Root) : 4 5 2 3 1**