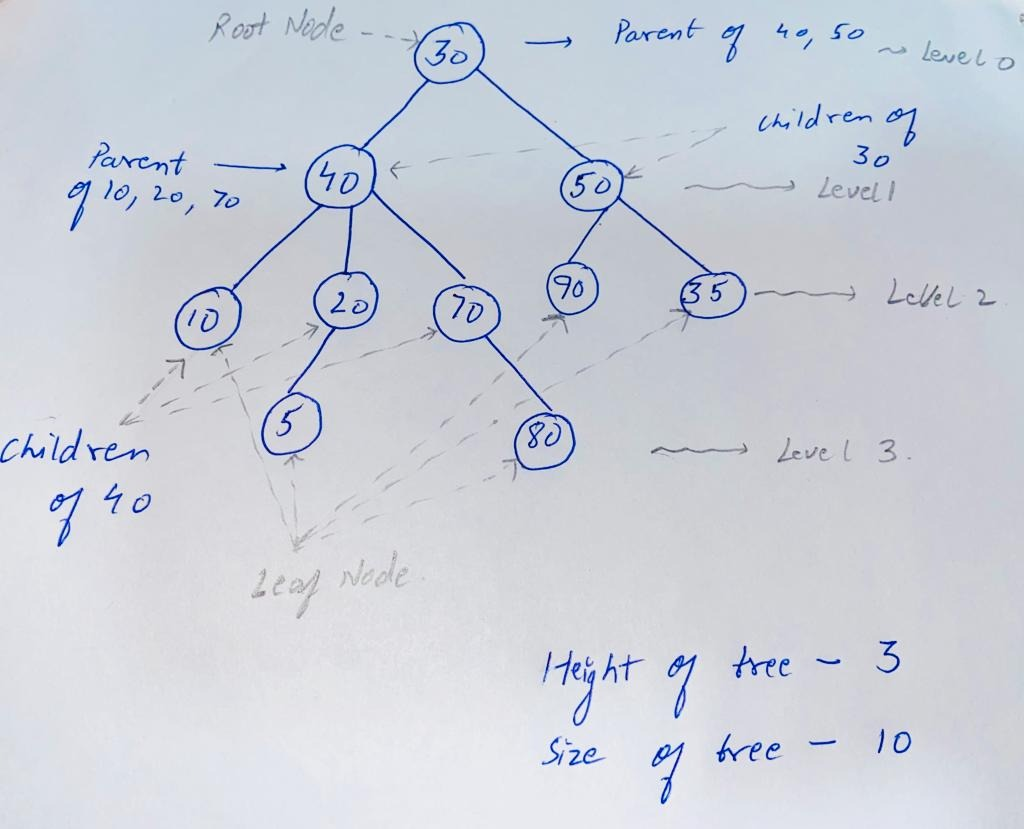
**Trees**

Trees is collection of nodes that are connected by edges and have a hierarchical relationship between the nodes, each node contains some data and two or more nodes having information about the their children. This data structure is used to organize and store data in such a way that we can access them more effectively.

Some of the terminologies used in Trees are:

1. Root Node – The first node of the tree from where the tree our tree is starting.
2. Leaf Node – The node which have 0 child.
3. Height of tree – Max distance of a node from the root node.
4. Size of tree – Total number of nodes present in the tree.



Trees are broadly categorize in three ways:

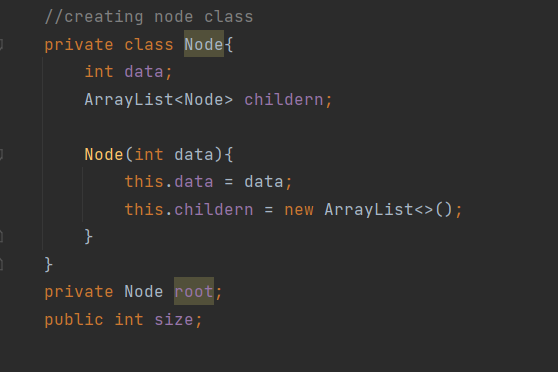
1. Generic Tree
2. Binary Tree
3. Binary Search Tree

Now let’s see what is the difference in all of these trees.

**Generic Trees**

In Generic Tree we don’t know how many children a particular node can have that means a particular node can have any number of children(nodes).

As we don’t know how many children a particular node will be having so for the internal implementation of generic tree we use Array List of Nodes which will contain information about all the children Nodes of a particular node.



**Binary Tree and Binary Search Tree**

In Binary Tree or Binary Search Tree a particular node can have at most 2 children Nodes that means a node can have 0 child or 1 child or 2 children not more than 2.

Now the question arrises that if Binary tree and Binary Search tree both of them can have at most two nodes then what is the need of Binary search tree?

A Binary Tree is only said to be Binary search tree only if the left child value is less than the node and the right child value is greater than the node. This makes the searching, insertion and deletion possible in O(log n) in the case of balanced tree. { where n is the size of the tree }

In binary tree or binary search tree, a node can only have at most 2 children that is the left child and the right child so every node contains only data, information about the left child and information about the right child.

