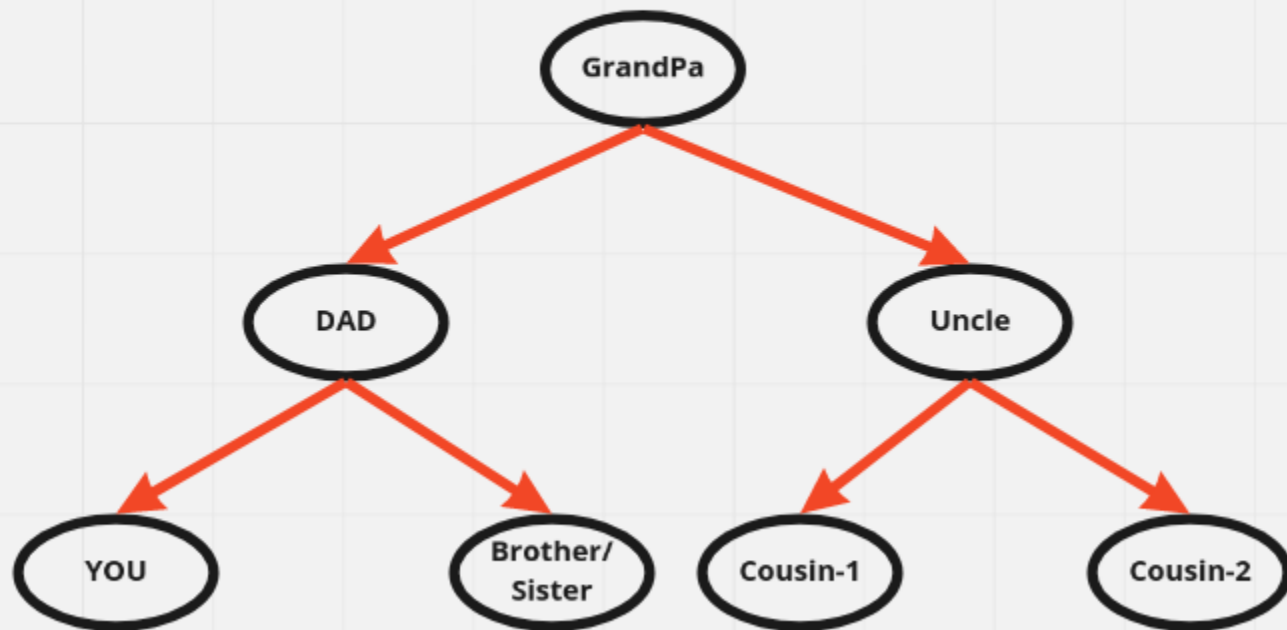
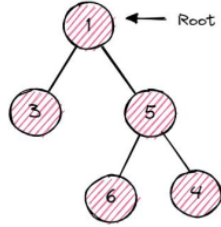


Trees - intro

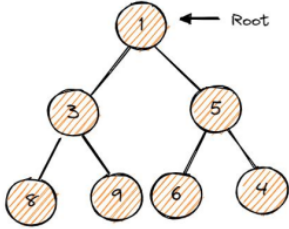
Family tree



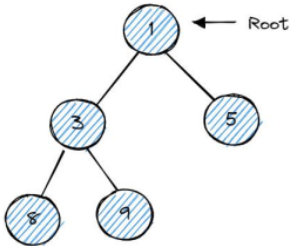
1. **Strict Binary Tree** :- A binary tree is called strict binary tree if each node has exactly two children or no children.



2. **Full Binary Tree** :- A binary tree in which each node have two children and all the leaf nodes are on the same level.

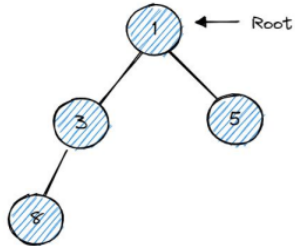


3. **Complete Binary Tree** :- A binary tree in which all the levels are completely filled except possibly the lowest one, which is filled from the left.



✓ Complete Binary tree

✓ Full Binary tree



✓ Complete Binary tree

✗ Full Binary tree

```

import java.util.*;
class Solution {
    class node {
        int data;
        node left;
        node right;
        public node(int d){
            this.data = d;
            this.left = null;
            this.right = null;
        }
    }

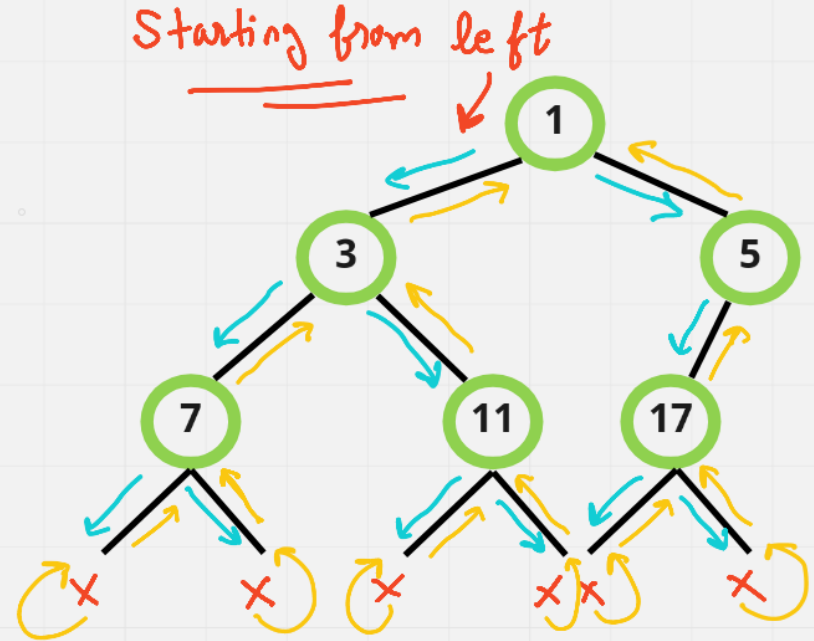
    public static node buildTree(node root){
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter the data: ");
        int data = scn.nextInt();

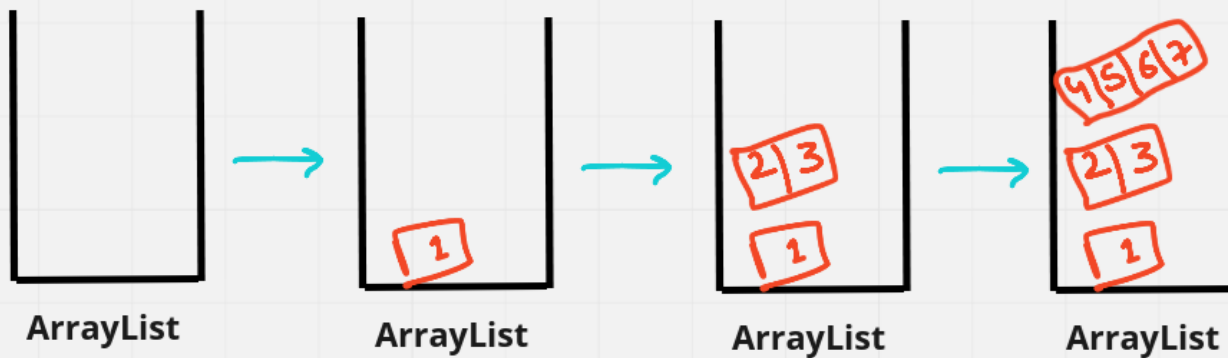
        root = new node(data);
        if(data == -1){
            return null;
        }

        System.out.println("Enter data for insterting in left " + data);
        root.left = buildTree(root.left);
        System.out.println("Enter data for insterting in right " + data);
        root.right = buildTree(root.right);
        return root;
    }

    public static void main(String[] args){
        node root = null;
        root = buildTree(root);
    }
}

```





```
class Solution {
    public List<List<Integer>> levelOrder(TreeNode root) {
        List<List<Integer>> res = new LinkedList<>();
        Queue<TreeNode> q = new LinkedList<>();

        if(root == null) return res;

        q.offer(root);

        while(!q.isEmpty()){
            int len = q.size();
            List<Integer> subres = new LinkedList<>();

            for(int i = 0; i < len; i++){
                if(q.peek().left != null) q.offer(q.peek().left);
                if(q.peek().right != null) q.offer(q.peek().right);

                subres.add(q.poll().val);
            }
            res.add(subres);
        }
        return res;
    }
}
```

Leetcode articles

<https://leetcode.com/discuss/study-guide/1820334/become-master-in-tree>