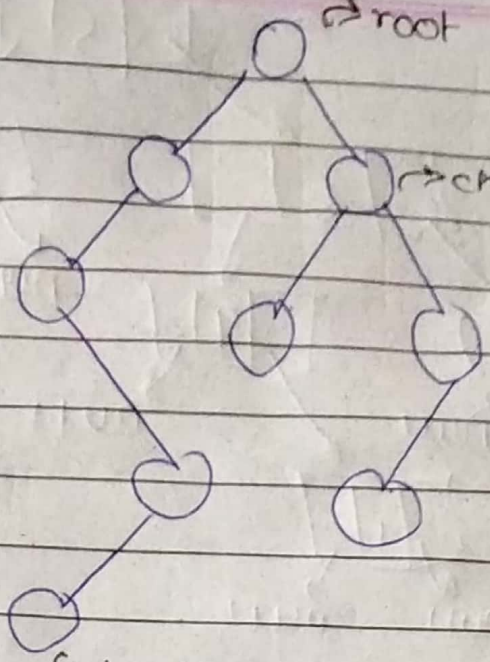
Trees



Sub-tree ⇒ Beneath guy of a node is subtree.

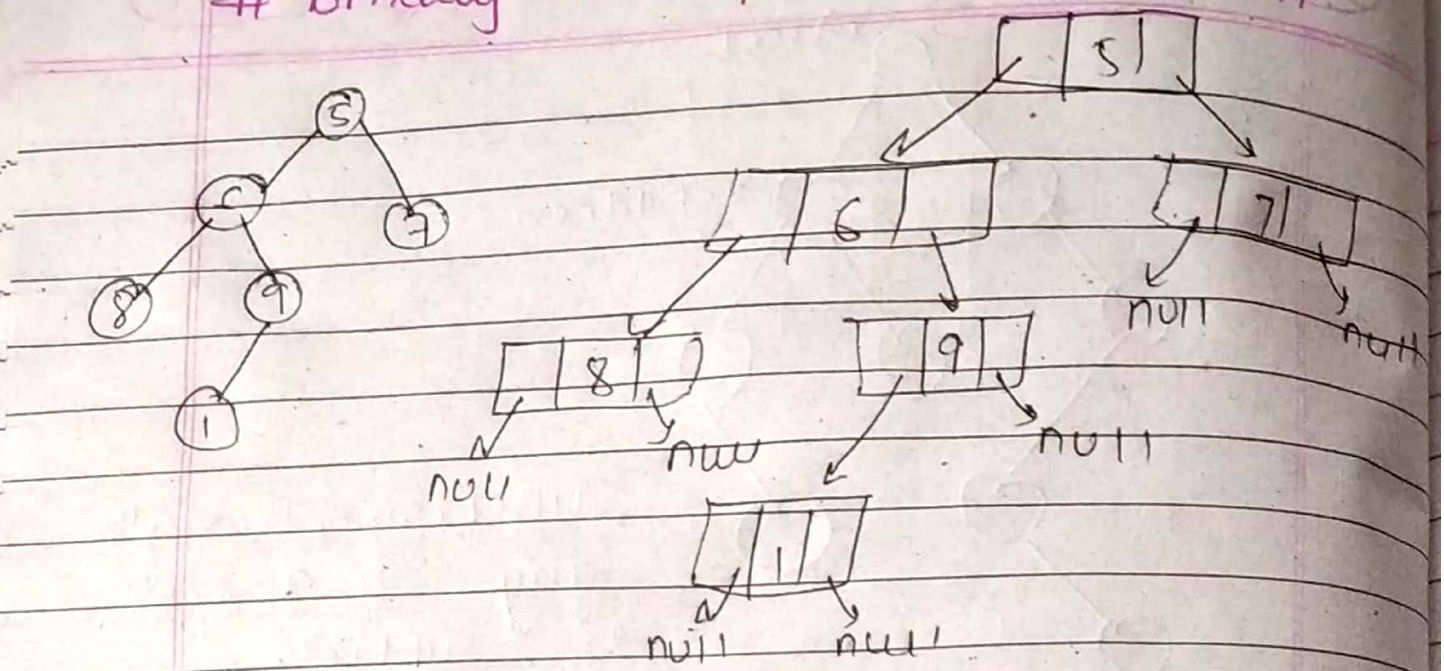
Ancestor ⇒ parent of a node, parent of parent of a node & so on are called as ancesstors

(Types of binary tree)

- (·) Full binary tree → either 0 or 2 children
- (·) Complete binary tree → all level should be completely filled except the last level ⇓ last level has all nodes on left as possible
- (·) perfect binary tree → all leaf nodes are at same level
- (·) Balanced binary tree
- (·) Degenerate tree

height ~~can be~~ of tree at max is $\log n$ (n → no. of nods)

Binary tree representation in cpp



(Implementation in cpp)

```
struct node {
    int data;
    struct Node *left;
    struct Node *right;

    Node (int val){
        data = val;
        left = right = null;
    }

};
int main(){
    struct Node *root = new Node(1);
    root -> left = new Node (2);
    root -> right = new Node(3);
    root -> left -> right = new Node (5);
}
```

→ If h = height of binary tree
maximum number of leaf nodes in
a binary tree = 2^h

maximum number of nodes in a
binary tree = $2^{h+1} - 1$

(implementation
in java)

```
public class binarySearchTree {
class node {
    int data;
    Node left, right;
    public Node(int data) {
        this.data = data;
        left = right = null;
    }
}
Node root = null;
public Node root;
publi
```