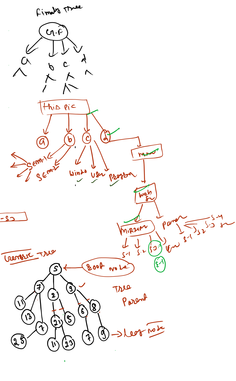
Tree

1. Binary Tree
2. Binary Search Tree
3. Height Balanced Tree

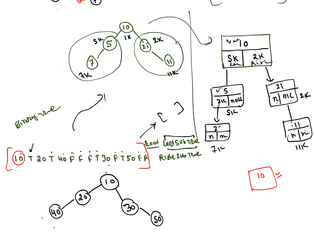
Binary Tree



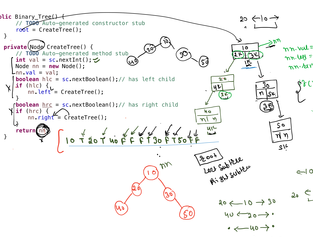
Binary Tree → B.T
Binary Search Tree → B.S.T

Left
Right

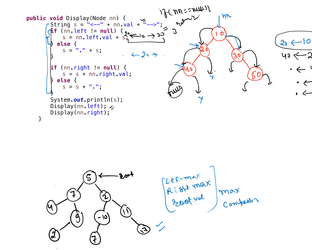
```
Class Node {
    int val;
    Node left;
    Node right;
}
```

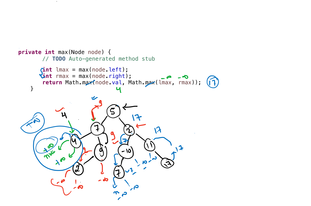



```
public Binary_Tree() {
    // TODO Auto-generated constructor stub
    root = CreateTree();
}

private Node CreateTree() {
    // TODO Auto-generated method stub
    int val = sc.nextInt();
    Node nn = new Node();
    nn.val = val;
    boolean hlc = sc.nextBoolean();// has left child
    if (hlc) {
        nn.left = CreateTree();
    }
    boolean hrc = sc.nextBoolean();// has right child
    if (hrc) {
        nn.right = CreateTree();
    }
    return nn;
}
```

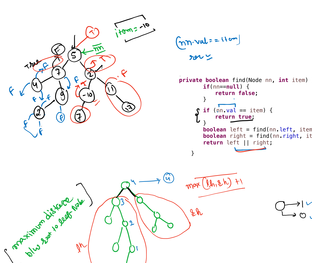


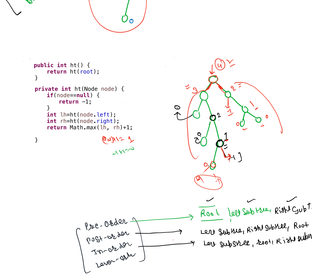

```
public void Display(Node nn) {
    String s = "";
    if (nn.left != null) {
        s = nn.left.val + " -->";
    } else {
        s = "." + s;
    }

    if (nn.right != null) {
        s = s + nn.right.val;
    } else {
        s = s + ".";
    }
    System.out.println(s);
    Display(nn.left);
    Display(nn.right);
}
```



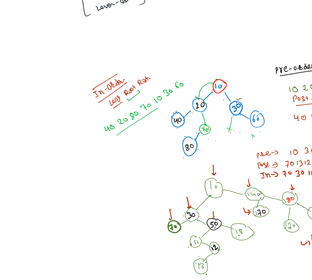



```
private int max(Node node) {
    // TODO Auto-generated method stub

    int lmax = max(node.left);
    int rmax = max(node.right);
    return Math.max(node.val, Math.max(lmax, rmax));
}
```






```
private boolean find(Node nn, int item) {
    if(nn==null) {
        return false;
    }

    if (nn.val == item) {
        return true;
    }
    boolean left = find(nn.left, item);
    boolean right = find(nn.right, item);
    return left || right;
}
```




```
public int ht() {
    return ht(root);
}

private int ht(Node node) {
    if(node==null) {
        return -1;
    }
    int lh=ht(node.left);
    int rh=ht(node.right);
    return Math.max(lh, rh)+1;
}
```

```
public void levelorder() {
    Queue<Node> q = new LinkedList<>();
    q.add(root);// add last
    while(!q.isEmpty()) {
        Node nnn=q.poll();// remove first;
        System.out.print(nnn.val+ " ");
        if(nn.left!=null) {
            q.add(nn.left);
        }
        if(nn.right!=null) {
            q.add(nn.right) ;
        }
    }
    System.out.println();

}
```