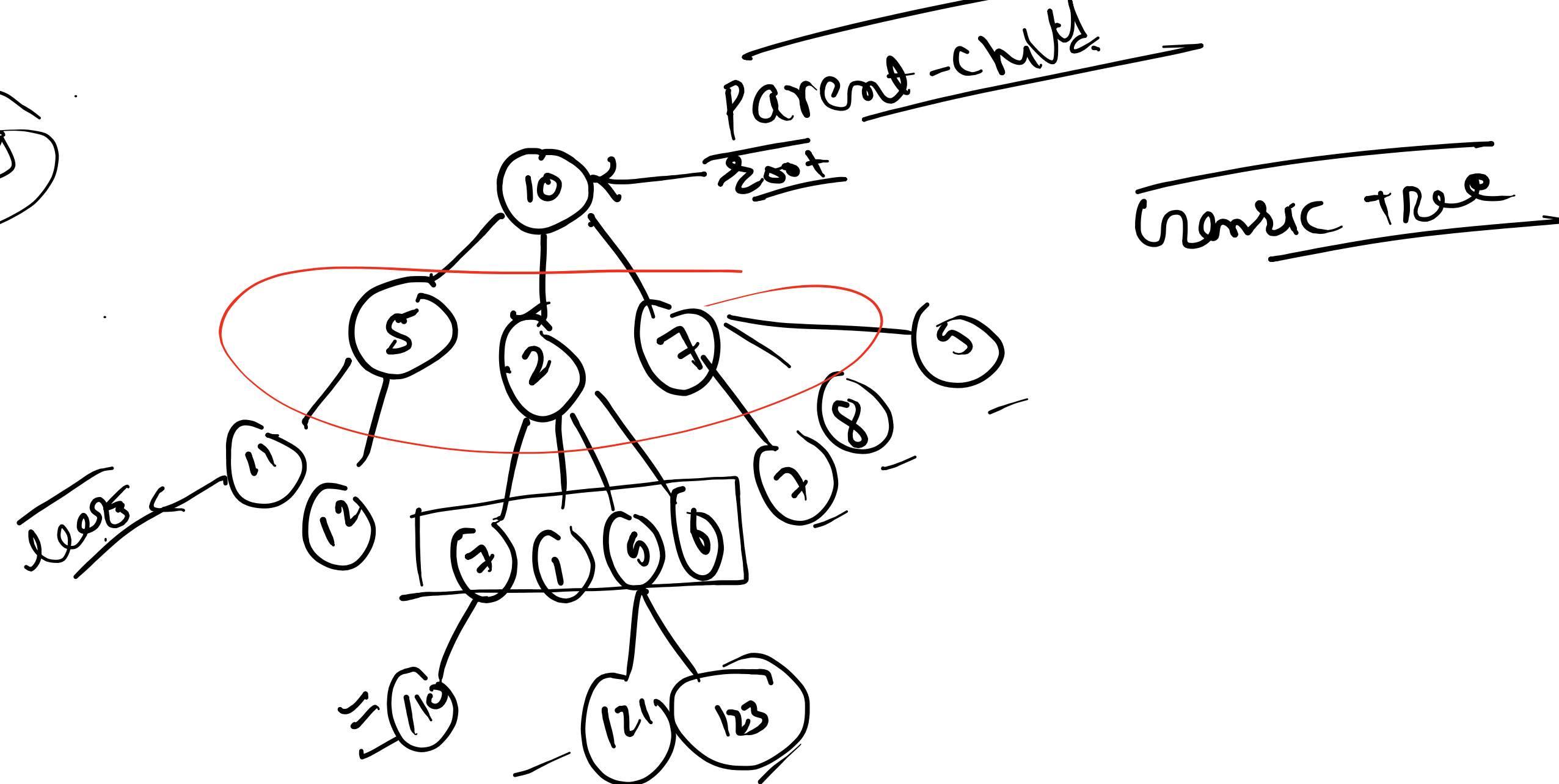
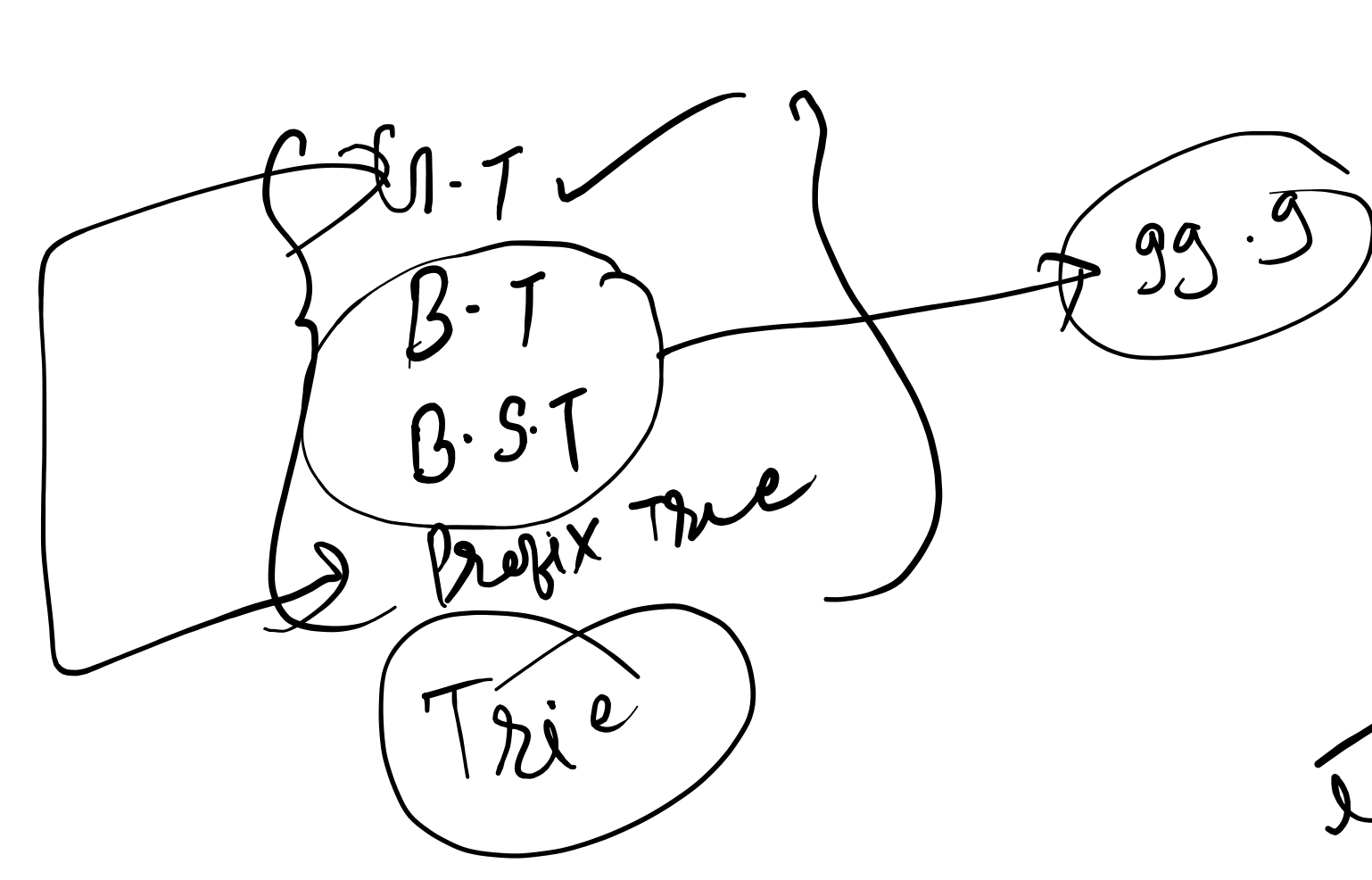
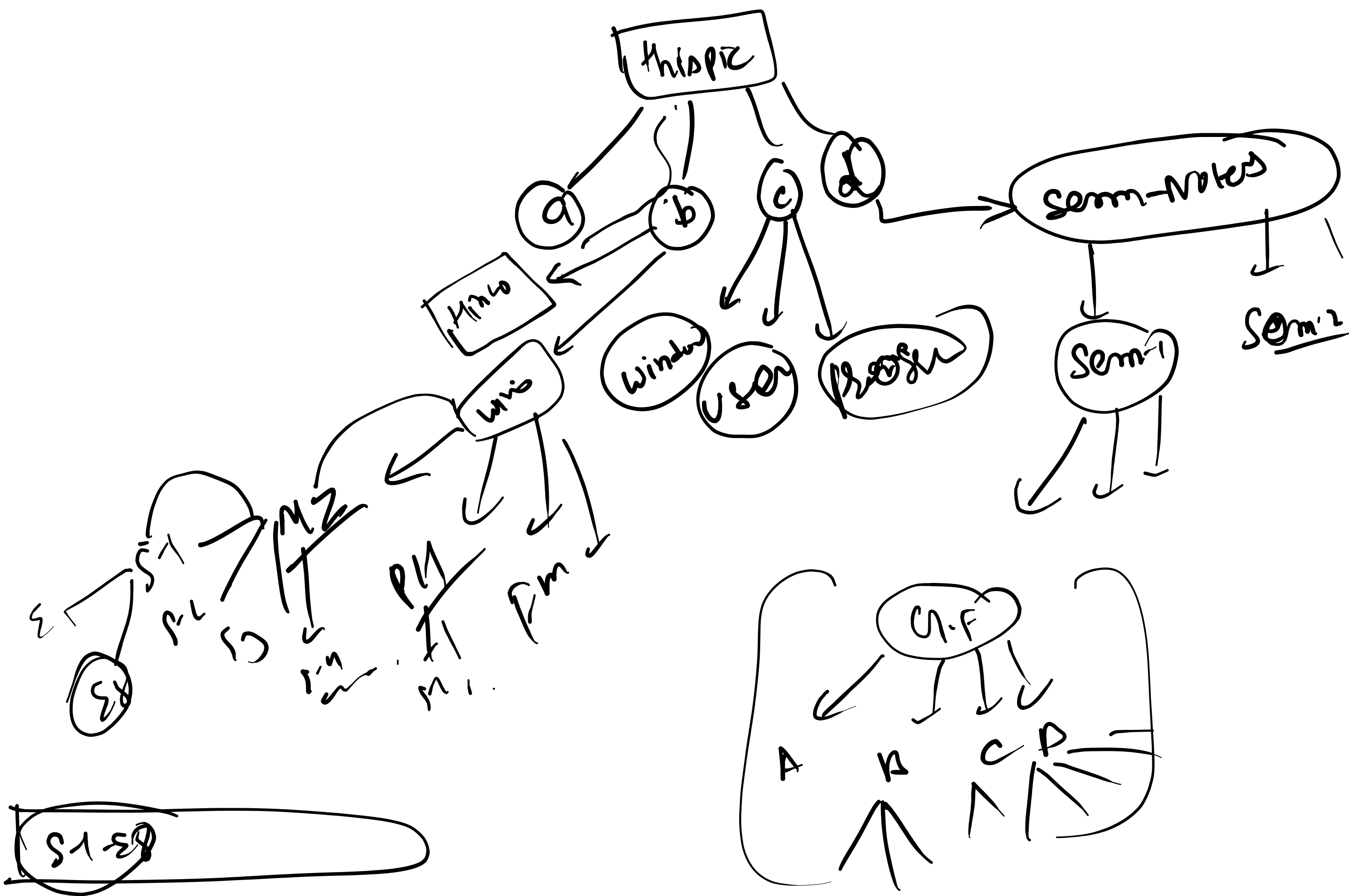


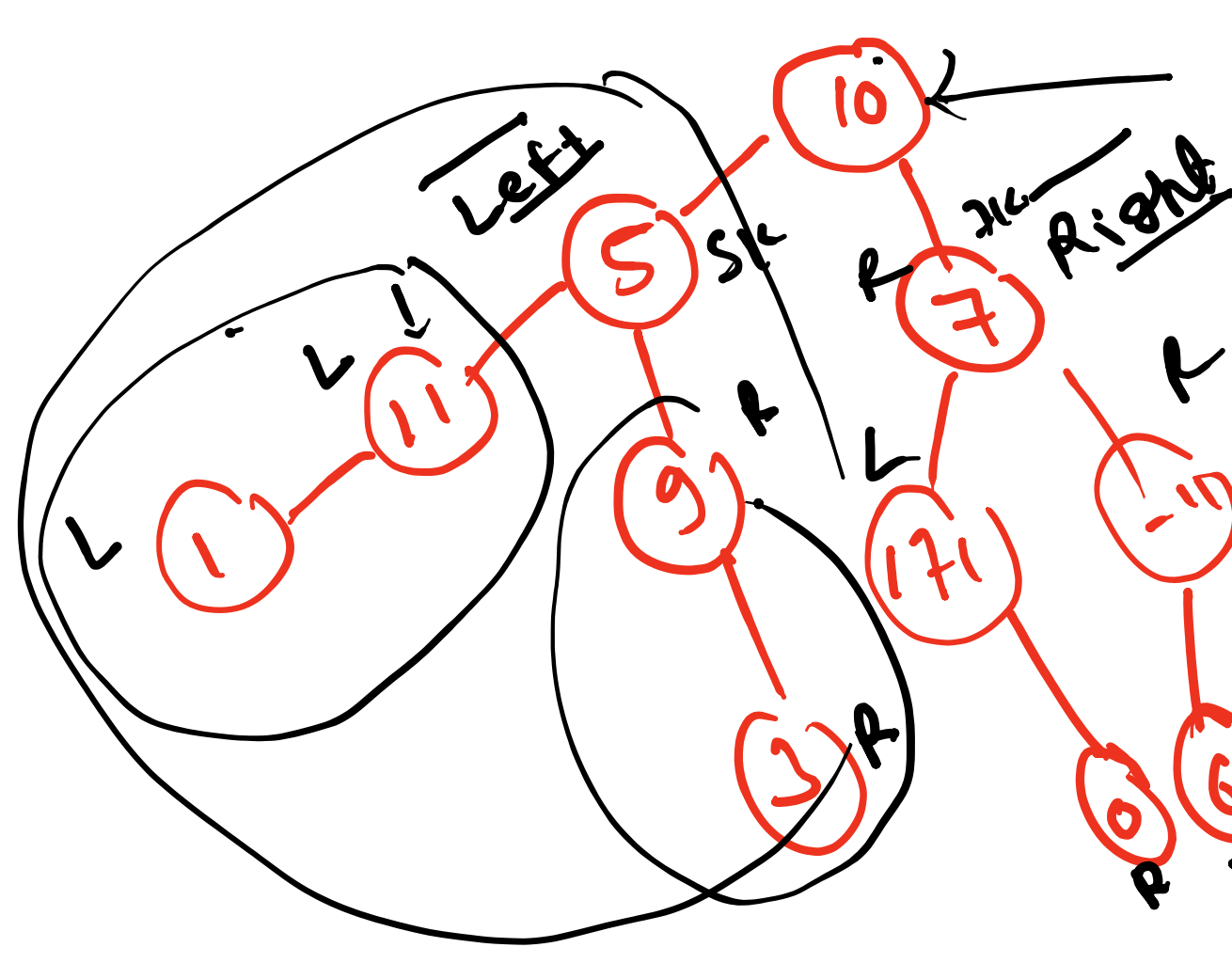
non-learn  
↳ Tree  
Use  
↳ File System



और वी और वी

B.T & B.S.T common → et max  
achieve

1000



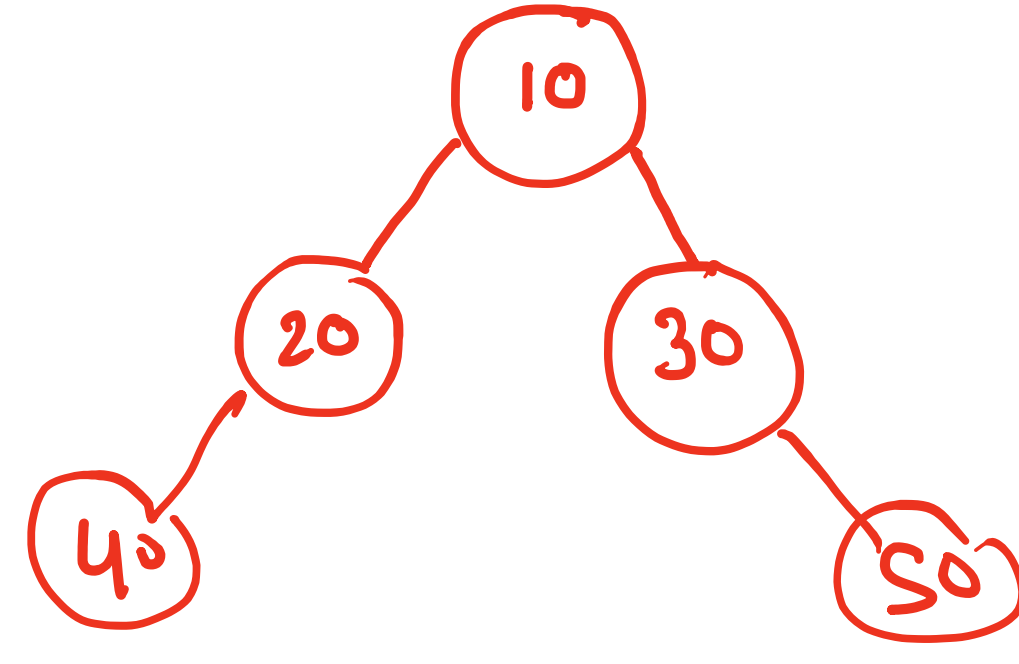
10-1

10  
2K 2K

Class Notes  
Node left  
Node right  
in val

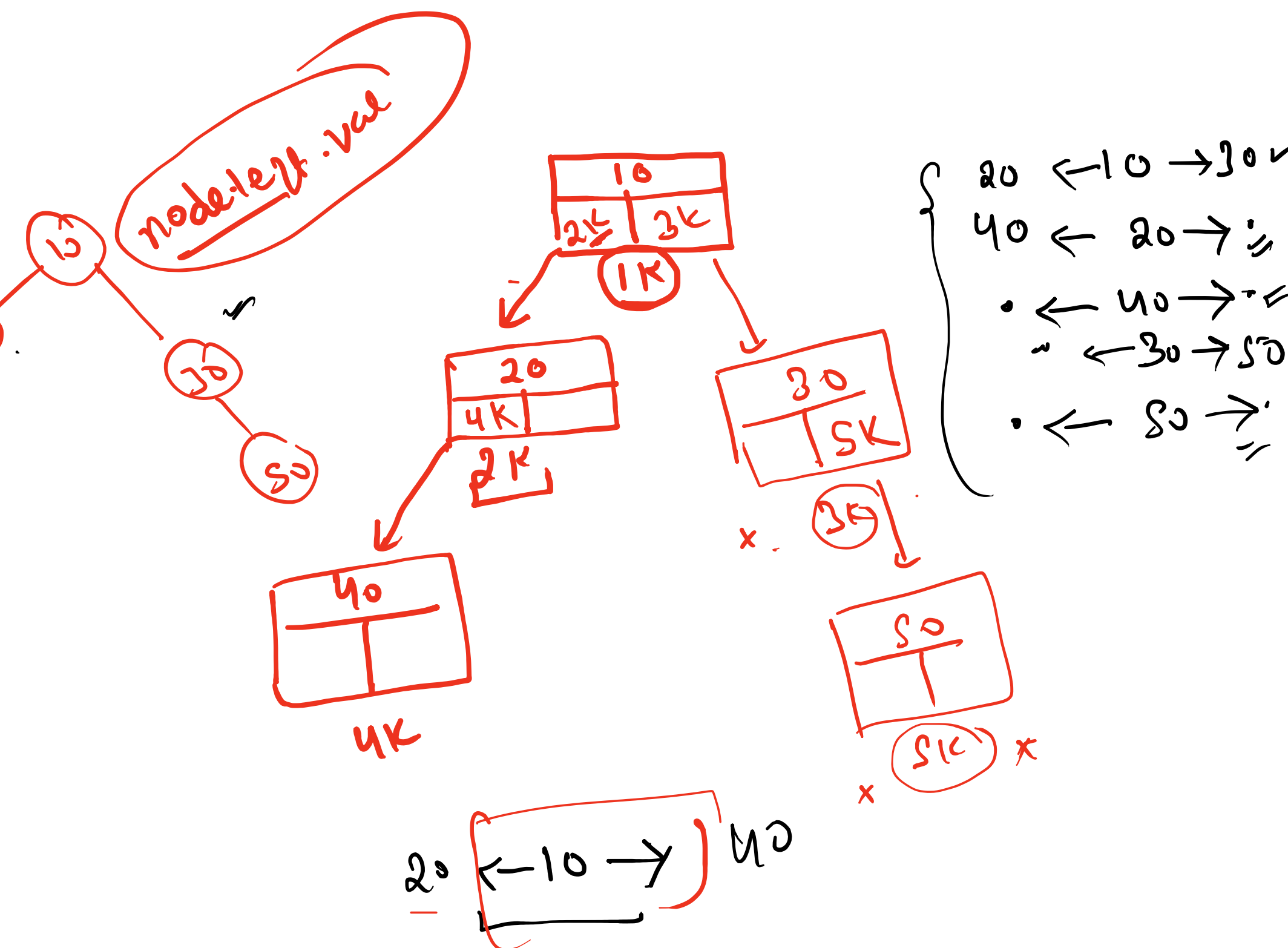
[ 10 T 20 T 40 F F T 30 F T 50 F F ] → insert

root, Left subtree, right subtree

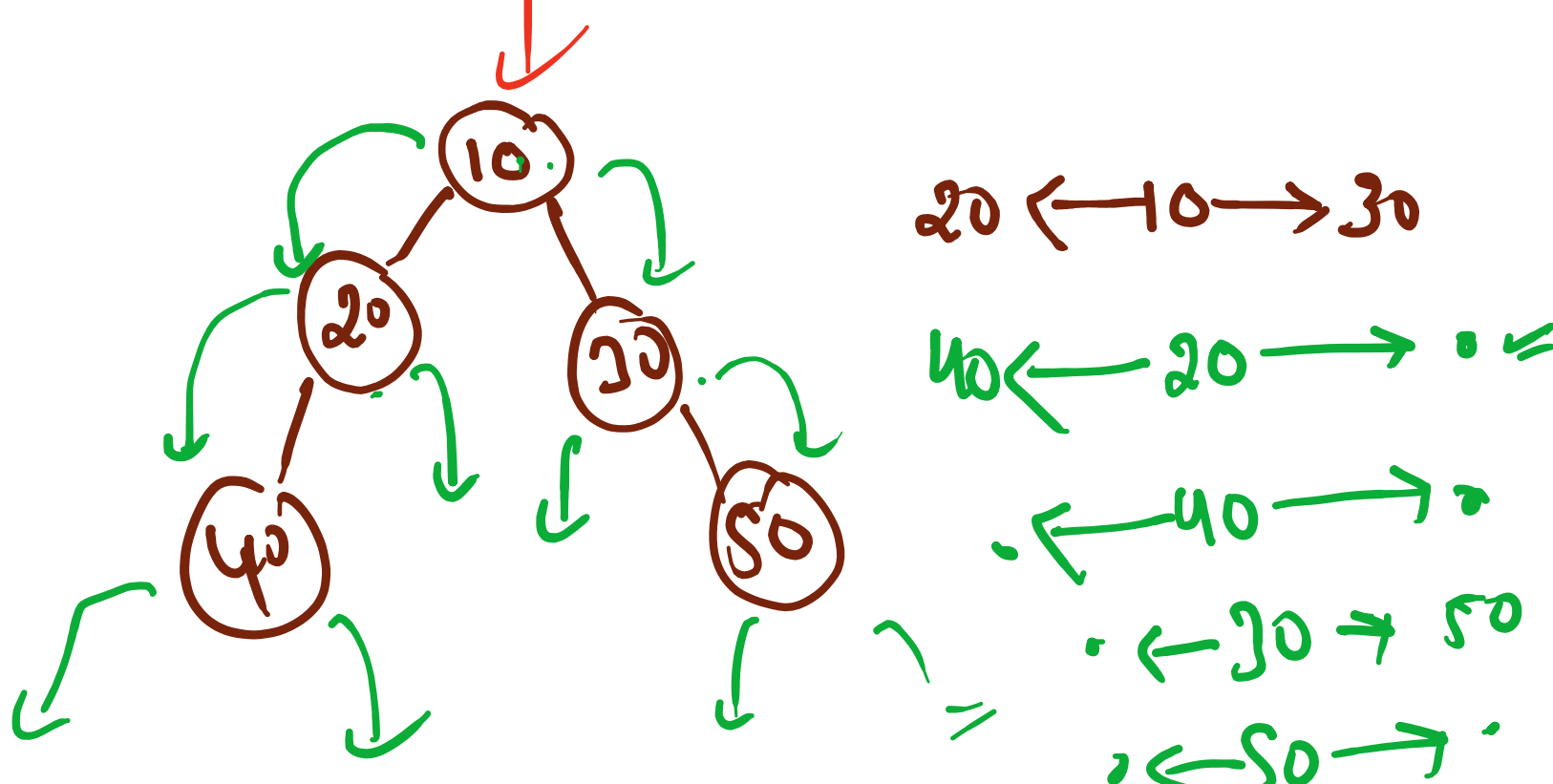


```
private Node Create_Tree() {
    // TODO Auto-generated method stub
    int val = sc.nextInt();
    Node nn = new Node(val);
    boolean hlc = sc.nextBoolean();
    if (hlc) {
        nn.left = Create_Tree();
    }
    boolean hrc = sc.nextBoolean();
    if (hrc) {
        nn.right = Create_Tree();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

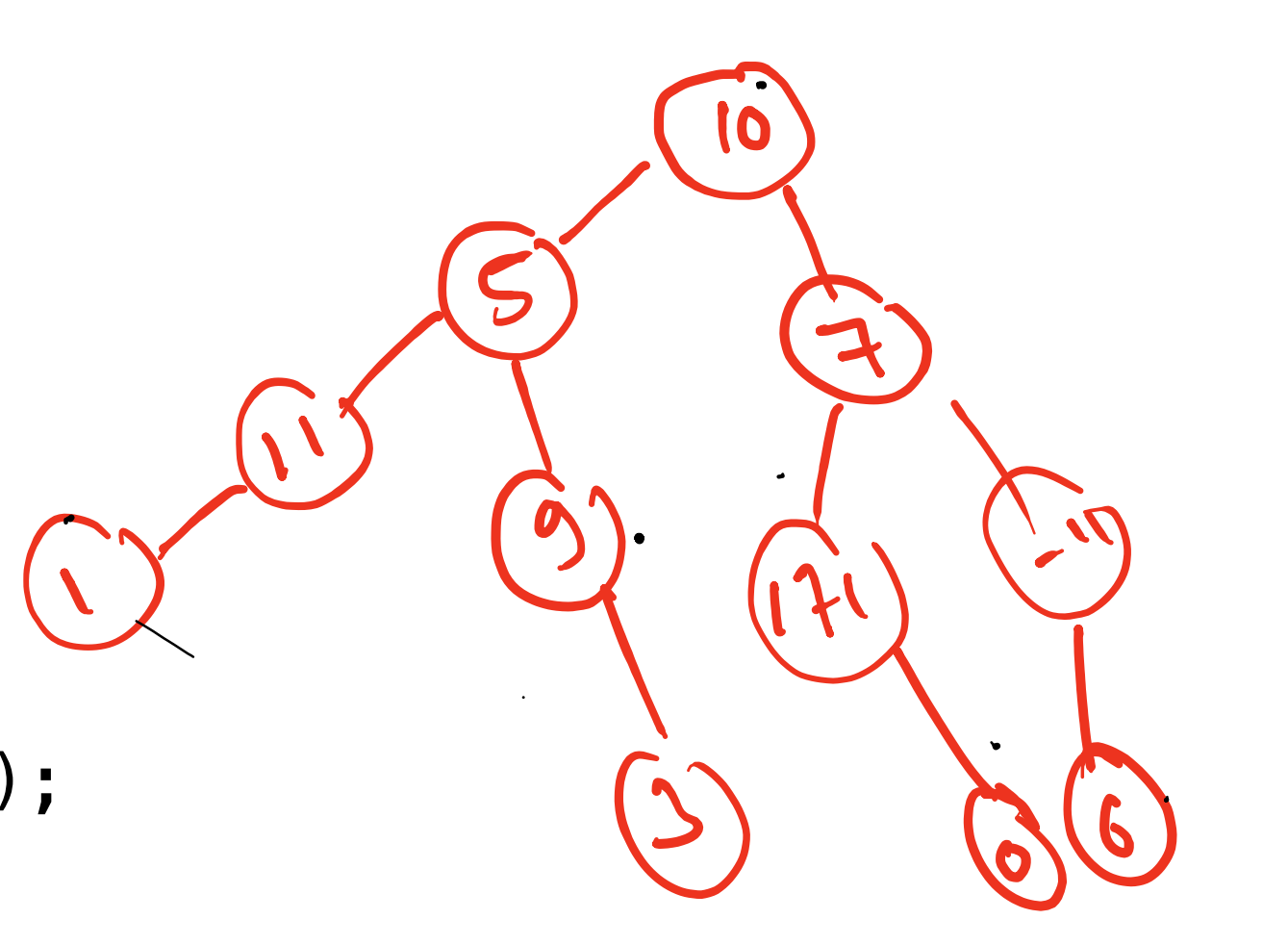
[ 10 T 20 T 40 F F T 30 F T 50 F F ]



```
public void Display(Node node) {
    String s = "<--" + node.val + "-->";
    if (node.left != null) {
        s = node.left.val + s;
    }
    else {
        s = "." + s;
    }
    if (node.right != null) {
        s = s + node.right.val;
    }
    else {
        s = s + ".";
    }
    System.out.println(s);
    Display(node.left);
    Display(node.right);
}
```

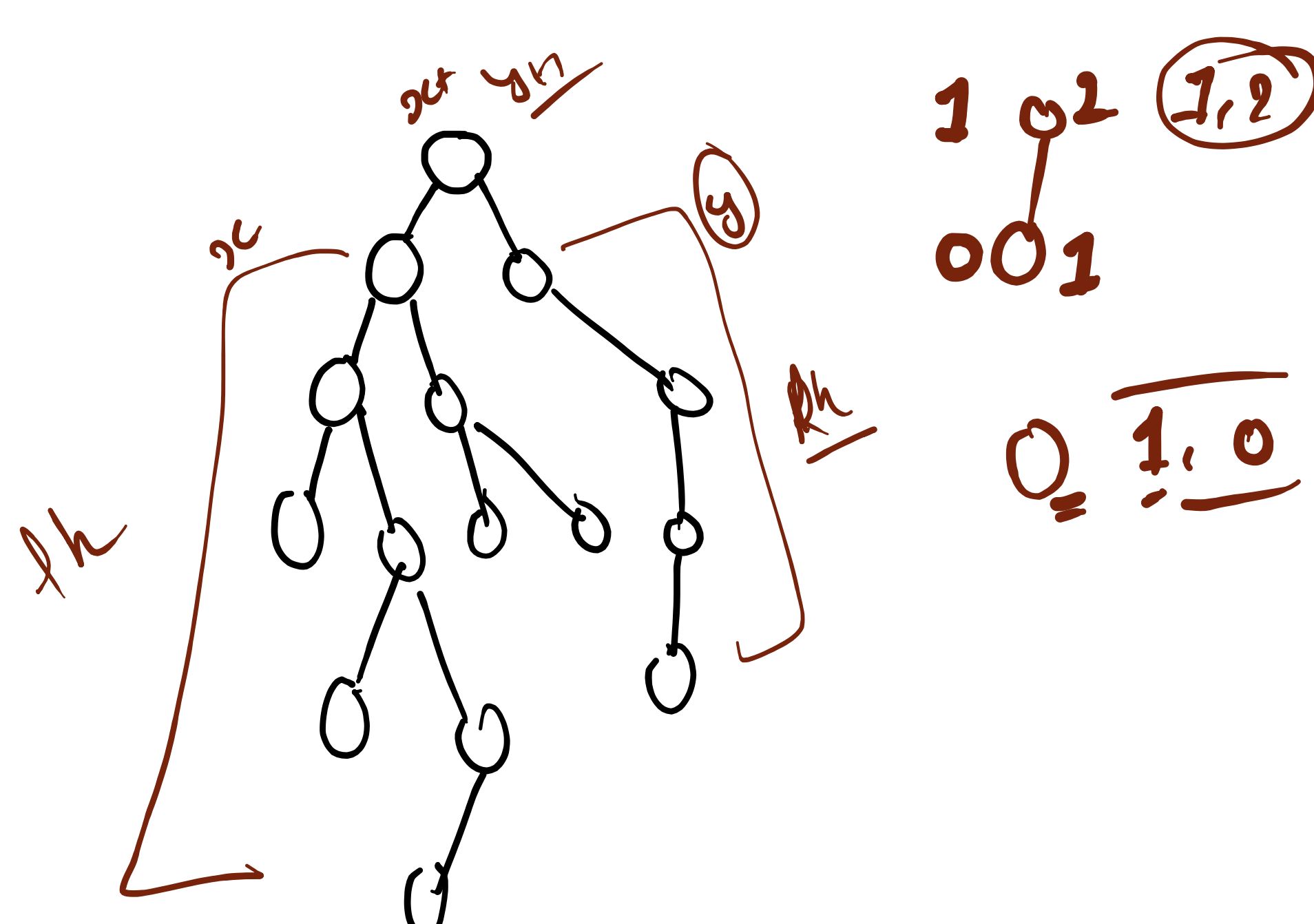


```
private boolean find(Node node, int item) {
    // TODO Auto-generated method stub
    if (node == null) {
        return false;
    }
    if (node.val == item) {
        return true;
    }
    boolean left = find(node.left, item);
    boolean right = find(node.right, item);
    return left || right;
}
```

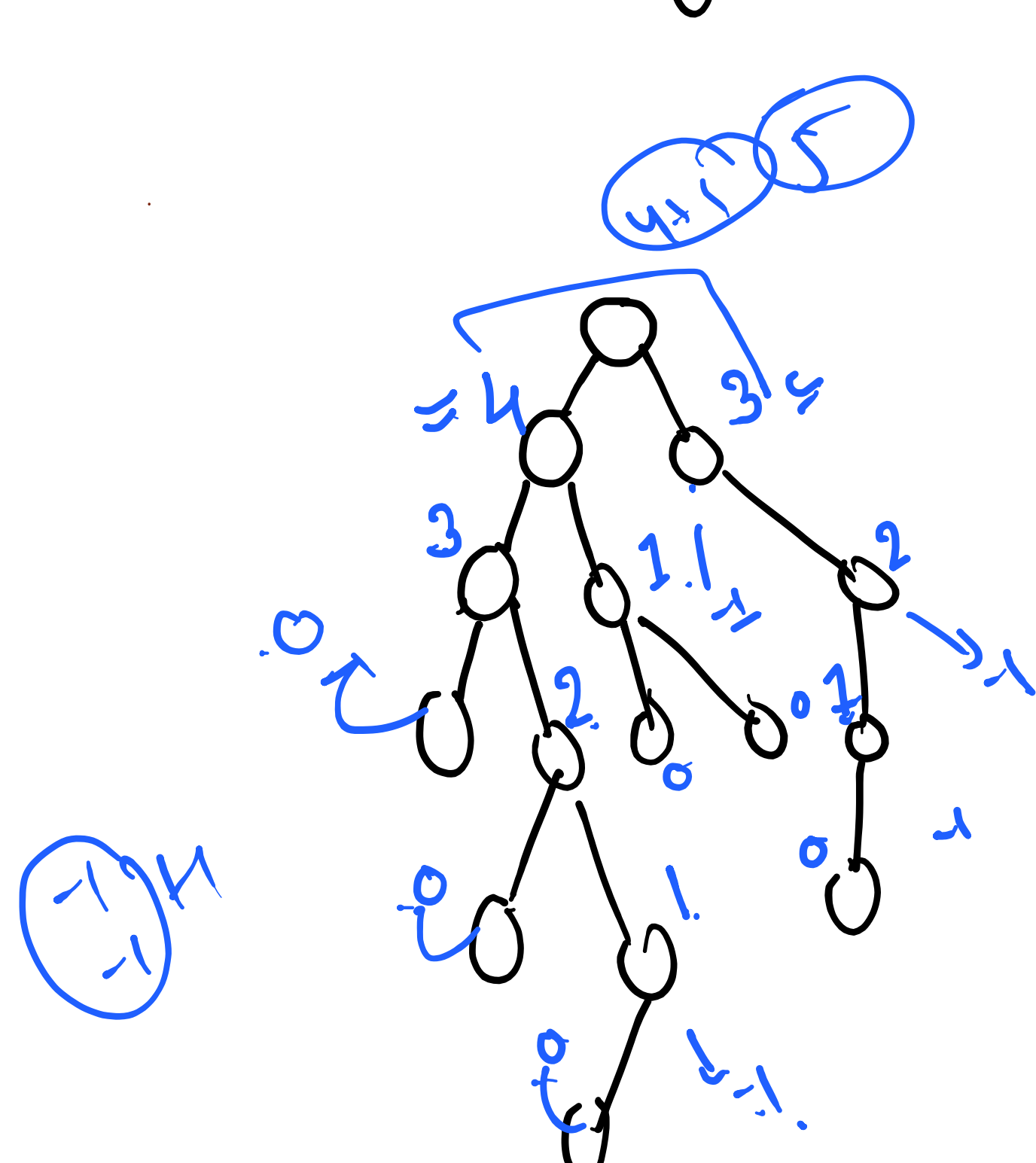


(-∞, ∞)

Maximum distance  
b/w root node & leaf node  
max(lh, rh) + 1

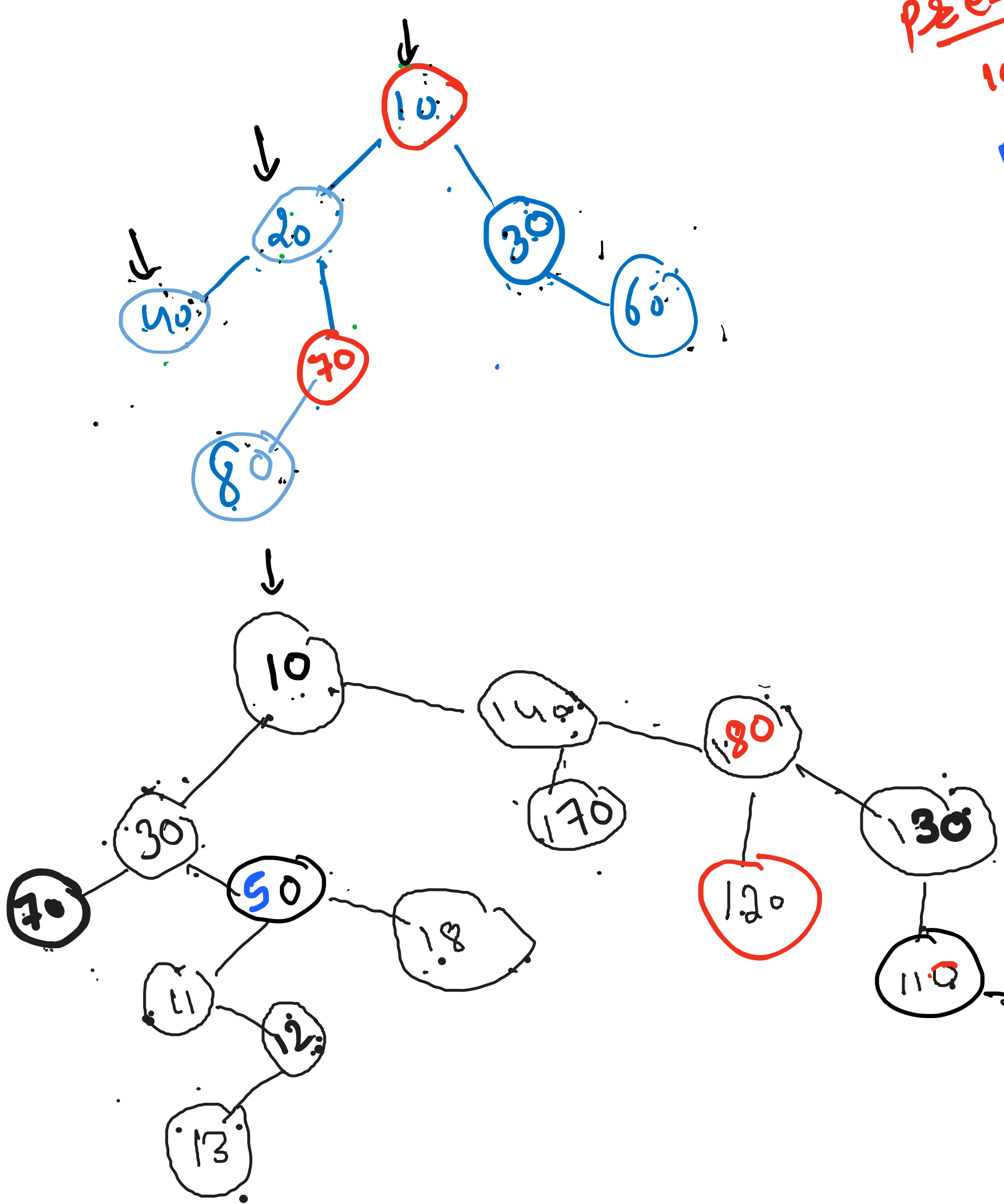


```
private int ht(Node node) {
    if (node == null) {
        return -1;
    }
    int lh = ht(node.left);
    int rh = ht(node.right);
    return Math.max(lh, rh) + 1;
}
```



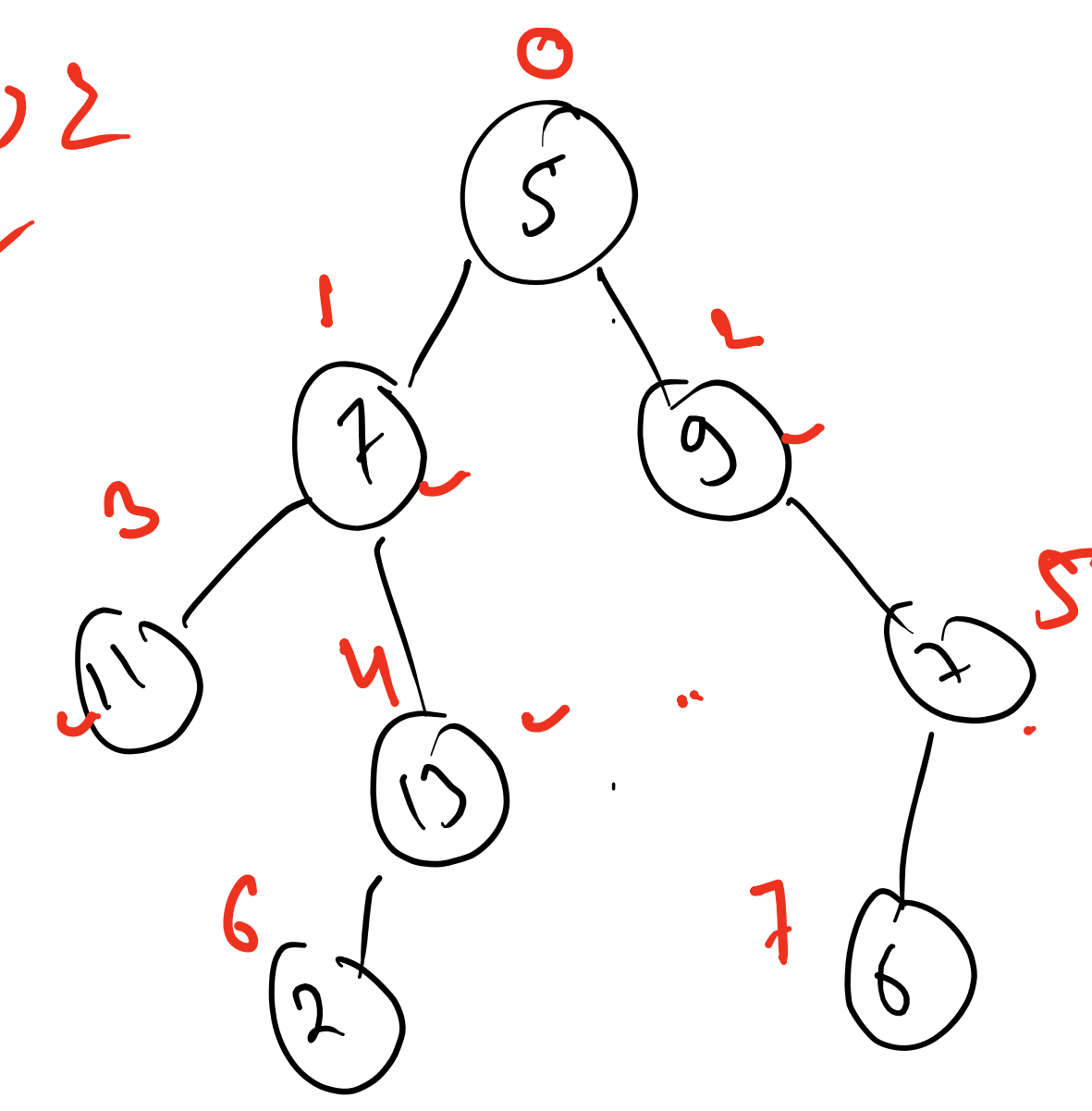
Pre-order  
Post-order  
In-order  
Level-order

Pre-order Root/Left/Right  
Post-order Left/Right/Root  
In-order Left/Root/Right  
Level-order Left/Right/Root



Pre-order  
Post-order  
In-order  
Level-order

while(!q.isEmpty())  
Node z = q.poll()  
if(z.left != null) L  
if(z.right != null) R



5 7 9 11 13 7 2 6  
0 1 2 3 4 5 6 7

8 7 9 11 13 7 2 6