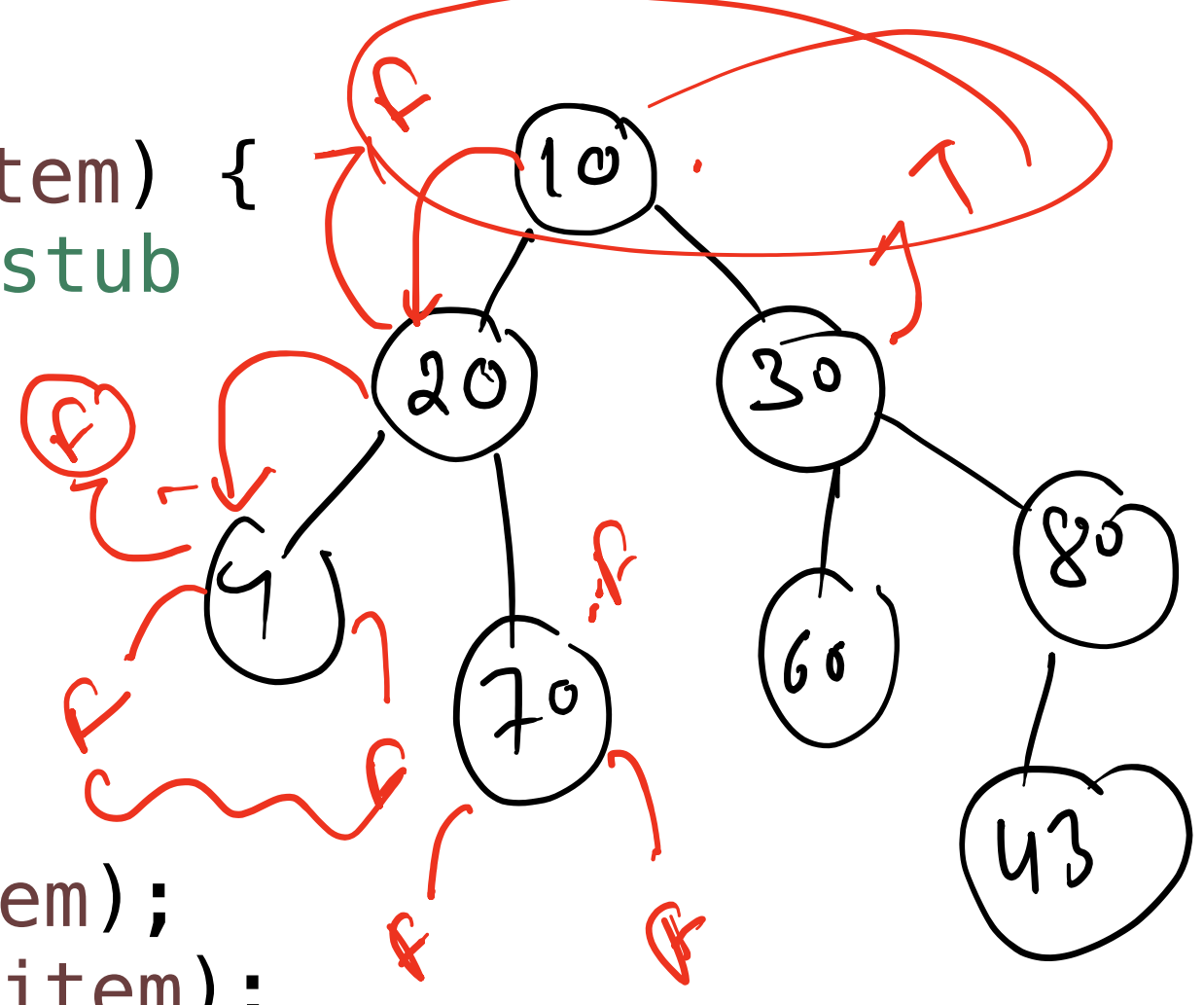
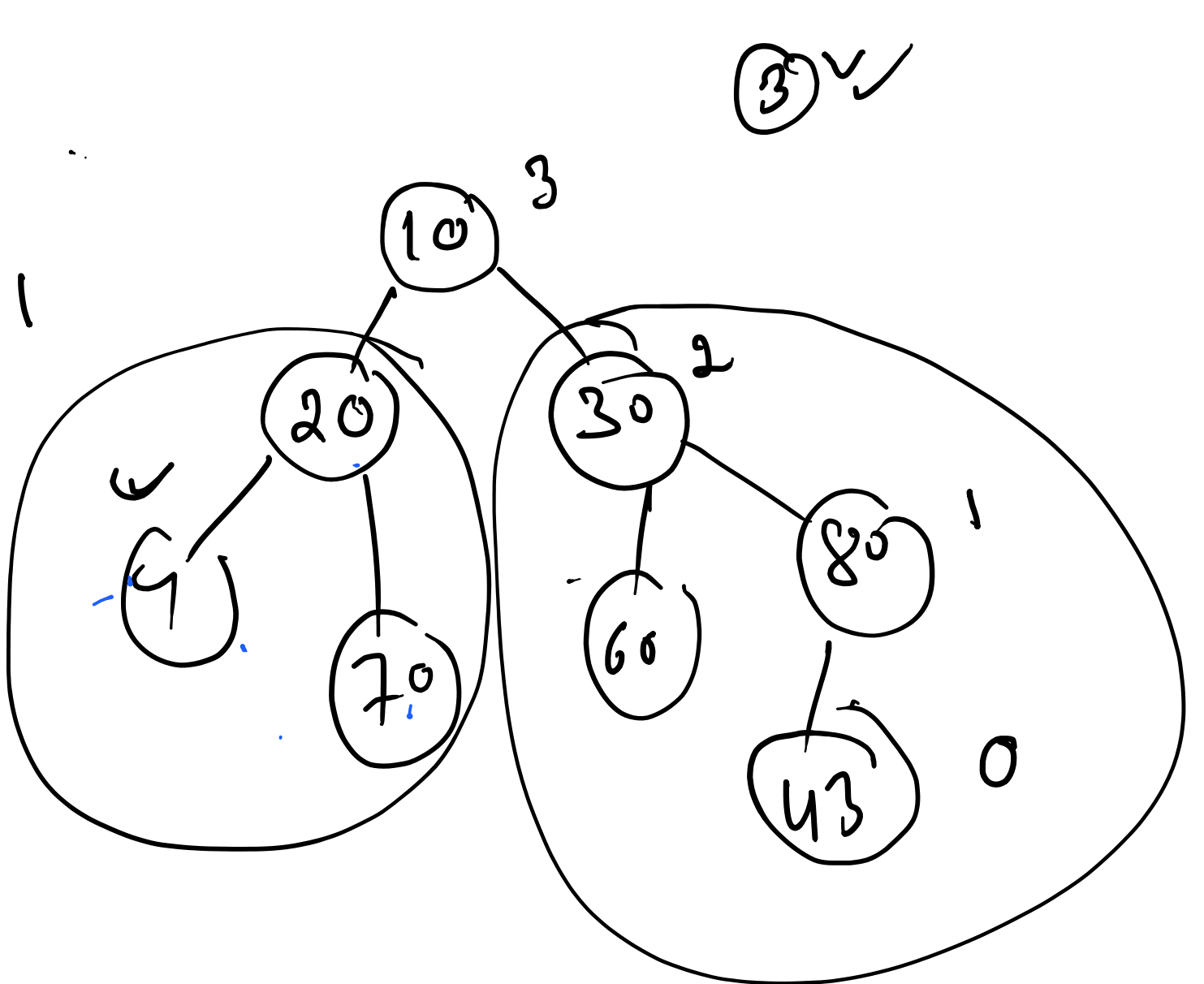
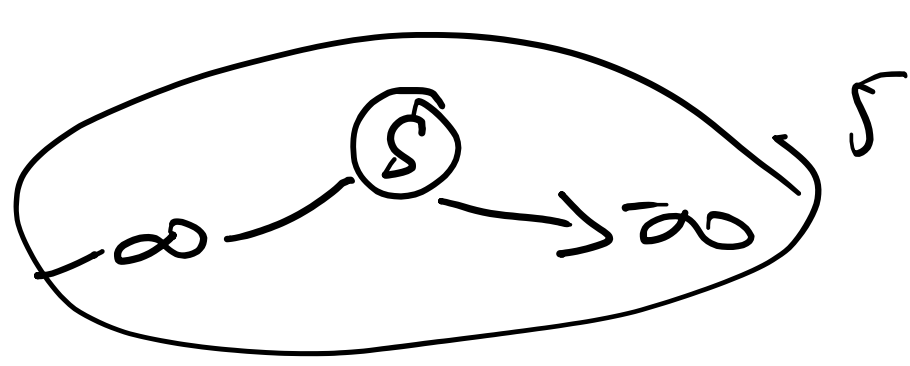
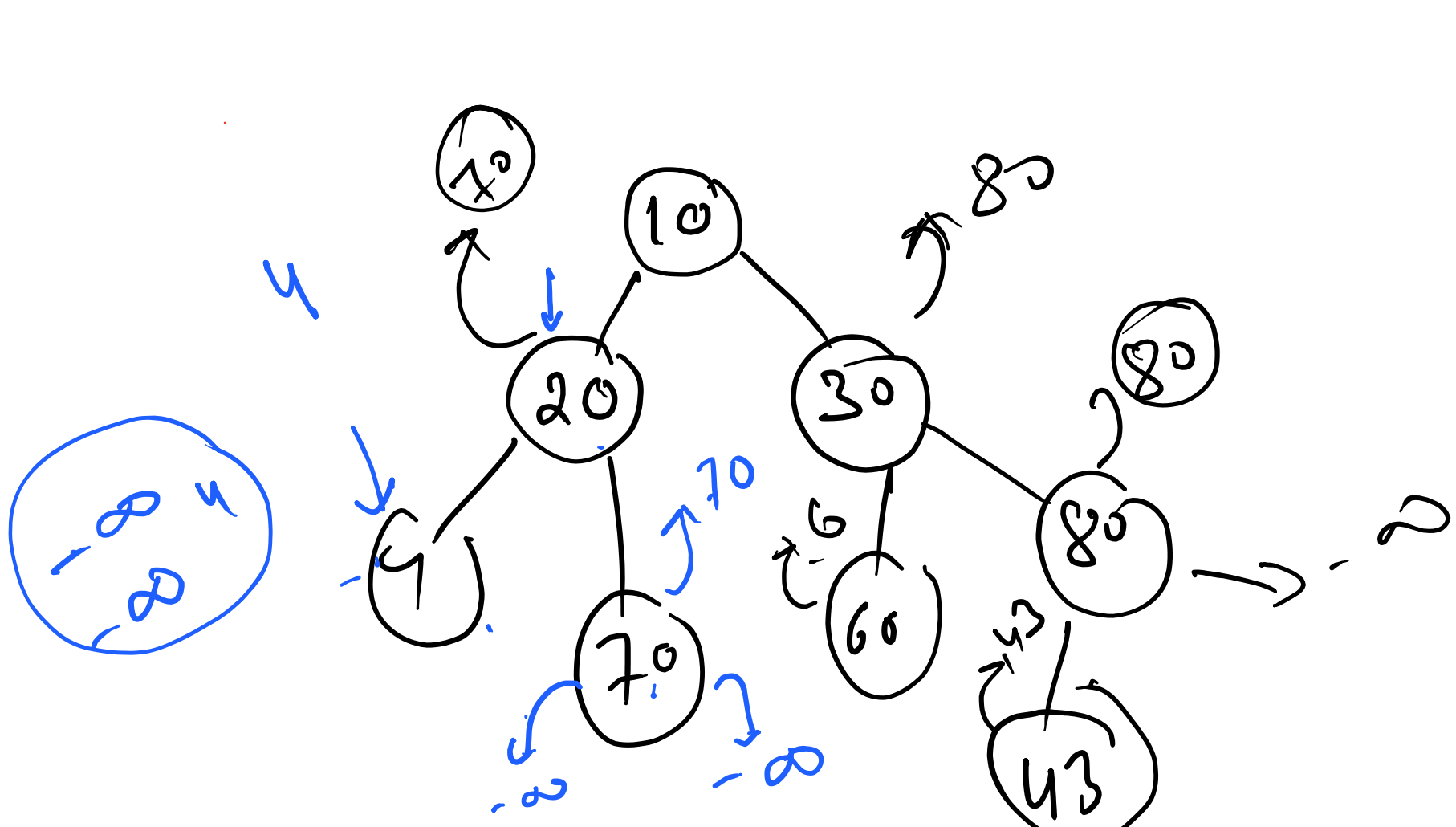


```
private boolean find(Node nn, int item) {  
    // TODO Auto-generated method stub  
    if(nn==null) {  
        return false;  
    }  
    if(nn.val==item) {  
        return true;  
    }  
    boolean left =find(nn.left, item);  
    boolean right =find(nn.right, item);  
    return left||right;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

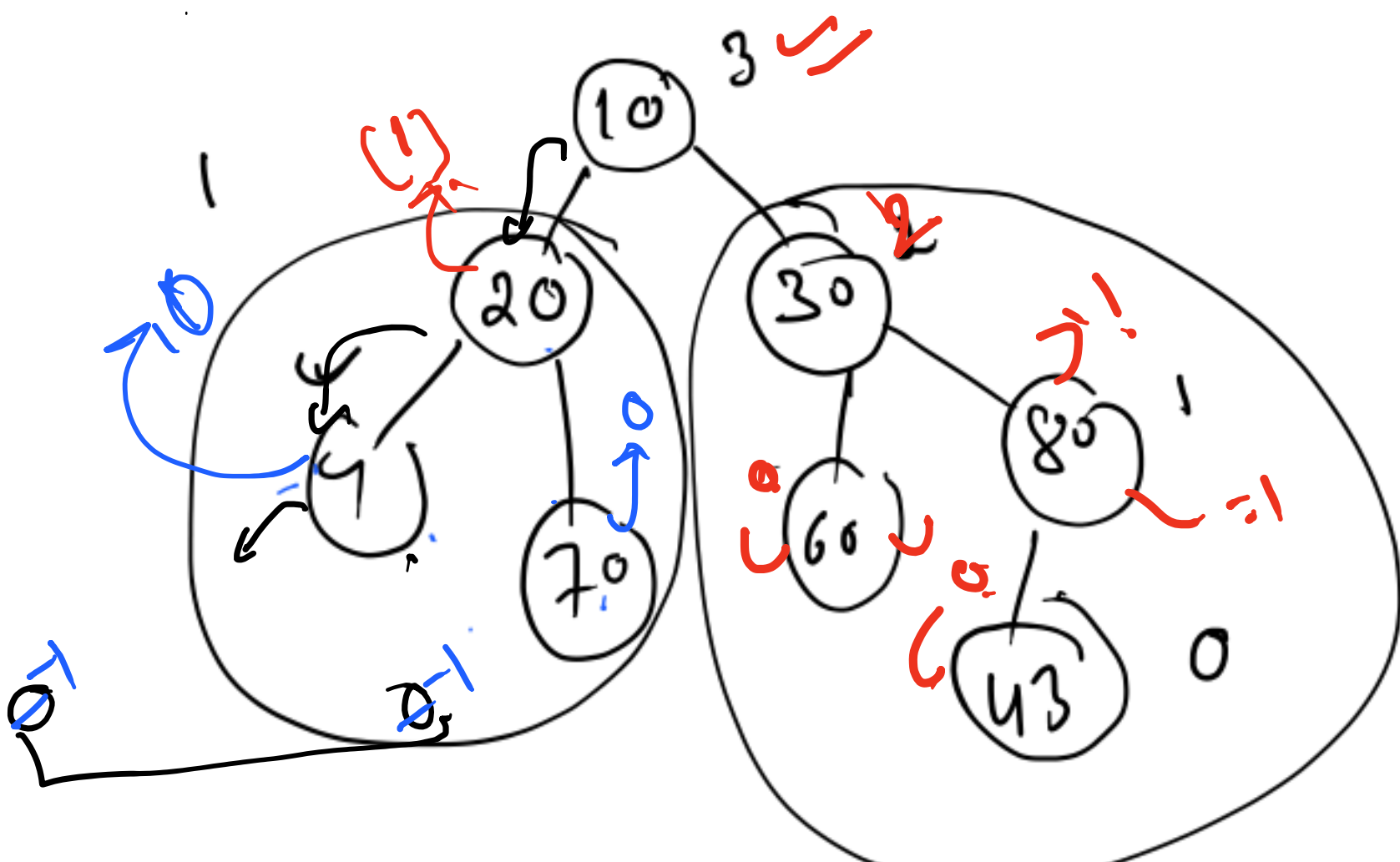


20



$$\frac{\textcircled{n} \rightarrow 0 \quad \textcircled{5} \neq 1}{(lh, rh) + 1}$$

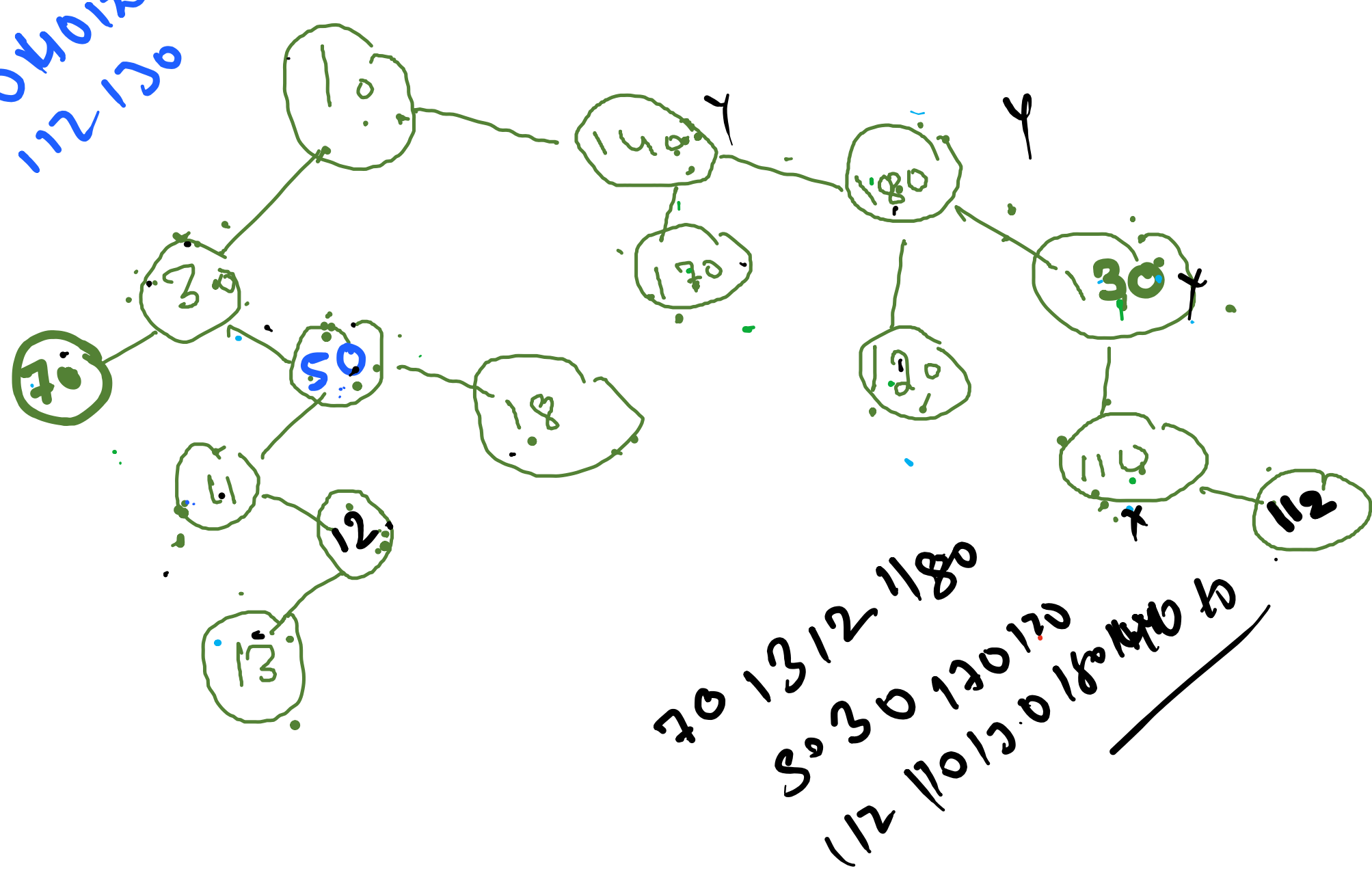
```
private int ht(Node nn) {  
    if(nn==null) {  
        return 0;  
    }  
    int lh=ht(nn.left);  
    int rh=ht(nn.right);  
    return Math.max(lh, rh)+1;  
}
```



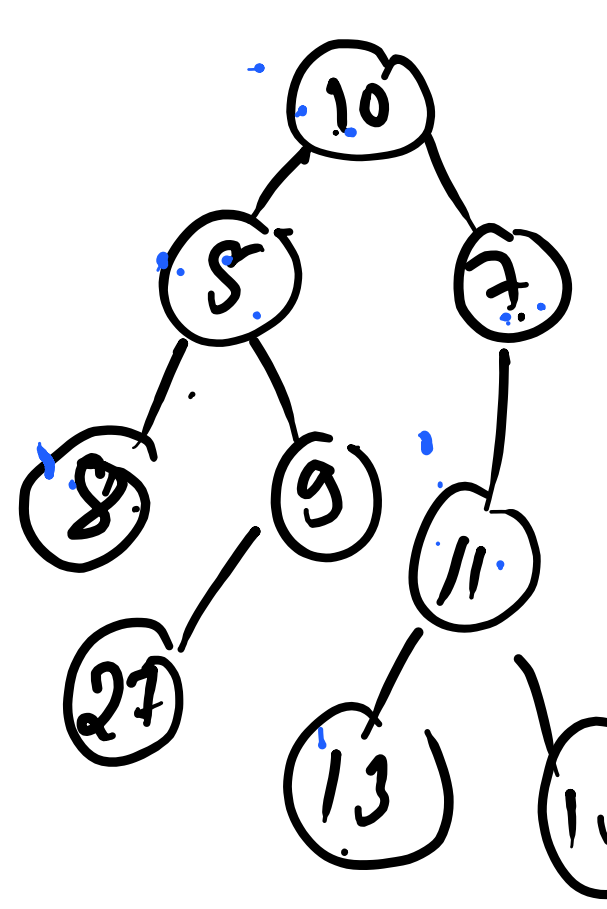
8 27 9 5 13 15 11 7 10

10 5 8 9 27 7 11 13
15
8 5 27 9 10 13 11 15 7

1min
70 30 11 13 12 80 18
10 170 60 100 180
110 112 130



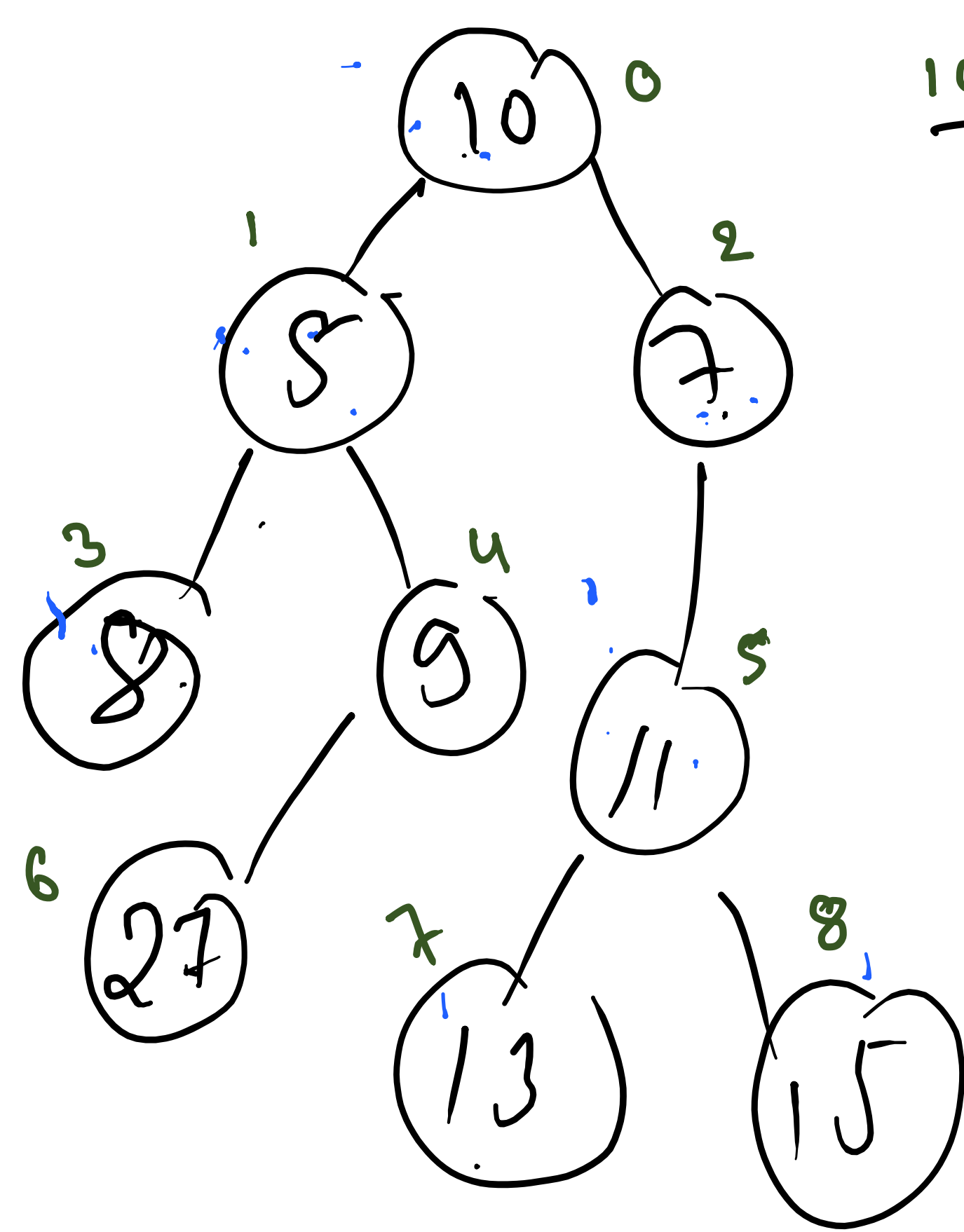
70 13 12 1180
50 30 130 170
112 110 130 160 140 150



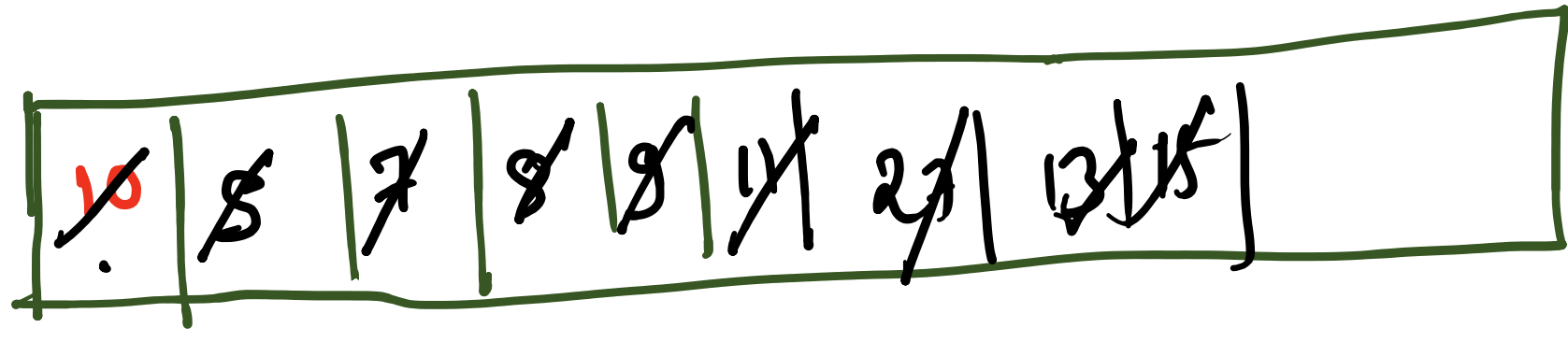
Per-order
In order
Post order

{ root Left Right
left root right
left right root }

10 30 70 50 11 12 13 18
140 170 180 120 130 110 112



10 5 7 8 9 11 27 13 15 10 5 7 8 9 11 27 13 15



```
public void LevelOrder() {  
    Queue<Node> q = new LinkedList<>();  
    q.add(root); // addlast  
    while (!q.isEmpty()) {  
        Node n = q.poll(); // q.remove() --> removefirst  
        System.out.print(n.val+" ");  
        if(n.left!=null) {  
            q.add(n.left);  
        }  
        if(n.right!=null) {  
            q.add(n.right);  
        }  
    }  
    System.out.println();  
}
```

