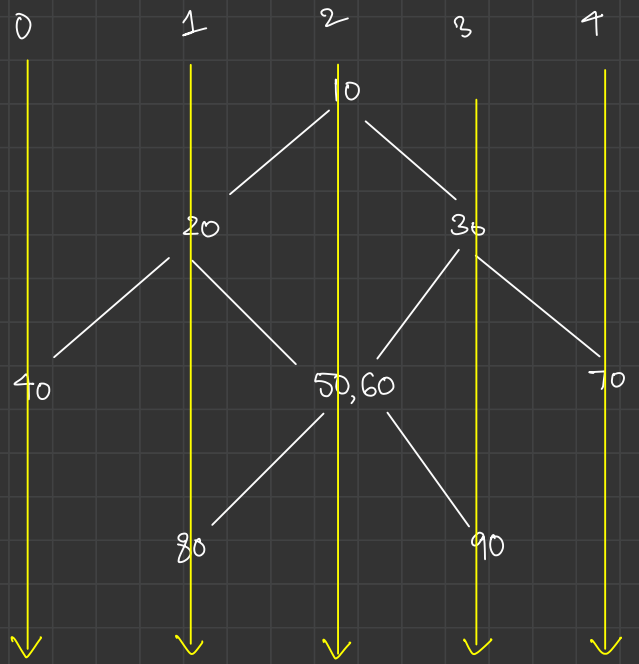
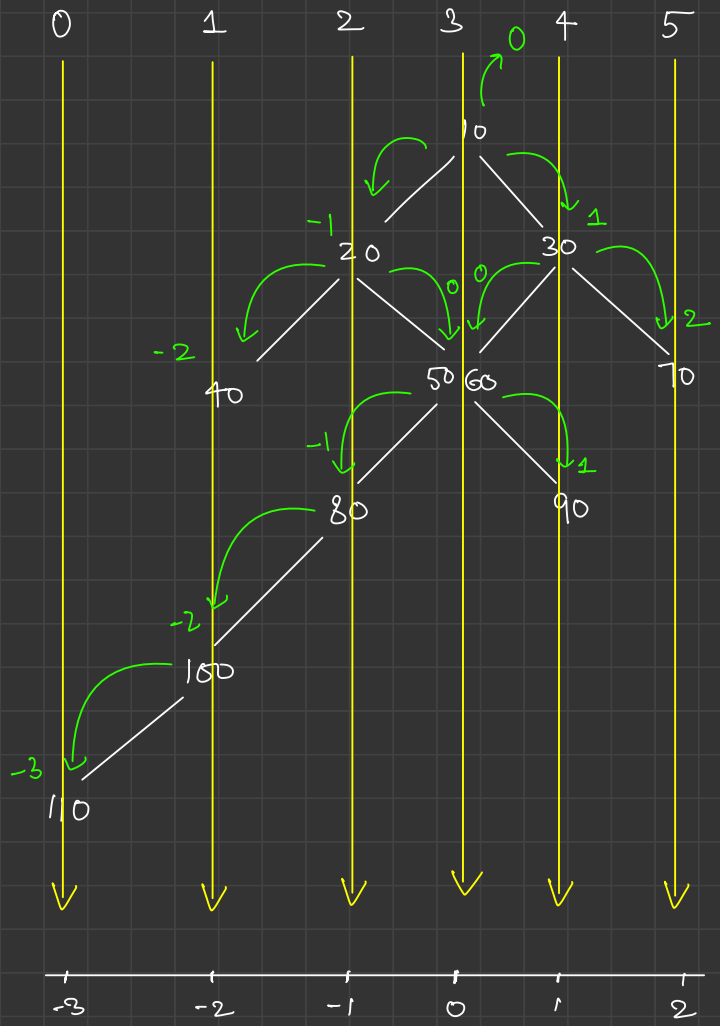
Vertical Order Traversal



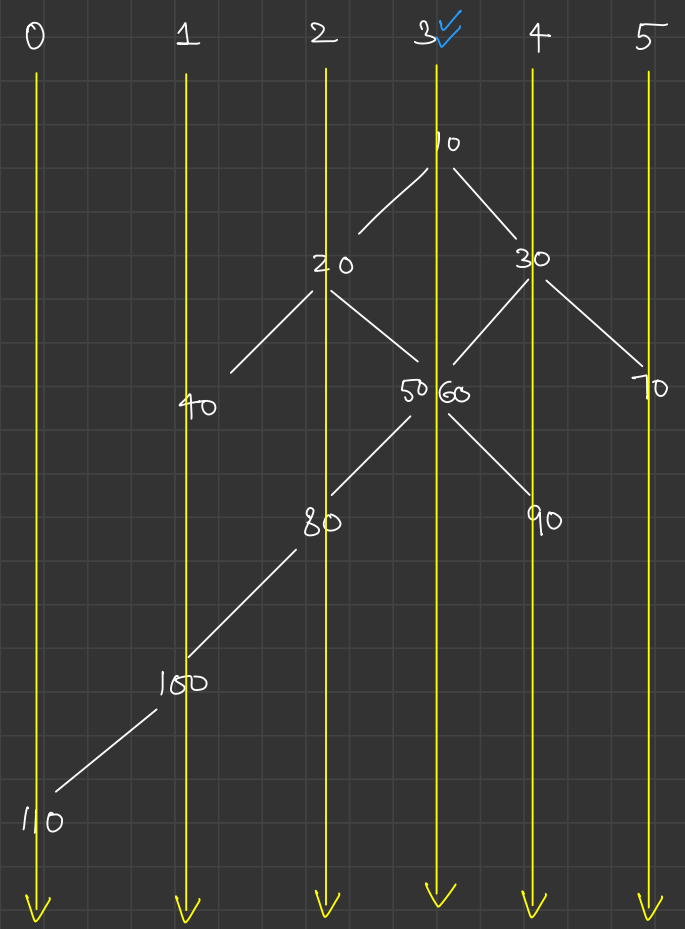
O/P

40
20, 80
10, 50, 60
30, 90
70



no. of levels = rightmost - leftmost + 1

root pos = - (leftmost)

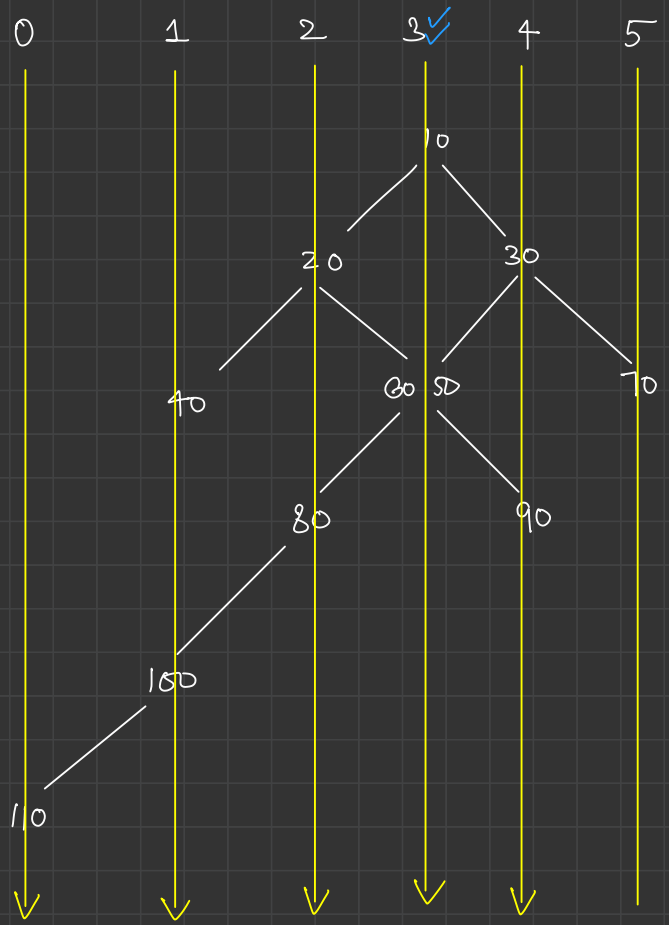


dp

0	110
1	40 , 100
2	20 , 80
3	10 , 50 , 60
4	30 , 90
5	70

(10,3) (20,2) (30,4) (40,1) (50,3) (60,3)

(70,5) (80,2) (90,4) (100,1) (110,0)



110
40 100
20 80
10 ~~60~~ ~~50~~ 50, 60
30 90
70

Priority Queue — 4th Module

sort

40, 60, 50, 70

remove leftmost node first
if vo is same, remove node with smallest value

```java
class Pair implements Comparable<Pair> {
    TreeNode node;
    int vl;

    Pair(TreeNode node, int vl) {
        this.node = node;
        this.vl = vl;
    }

    @Override
    public int compareTo(Pair o) {
        if (this.vl == o.vl) {
            // return smaller value person before
            return this.node.data - o.node.data;
        } else {
            // return person with smaller vl
            return this.vl - o.vl;
        }
    }
}
```

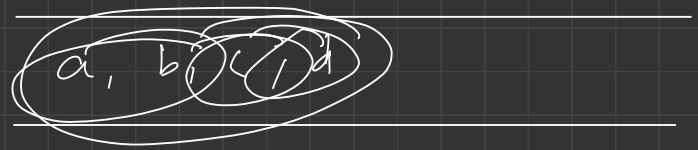
Comparable<Pair> → sorting → how to sort?

this, other

custom logic

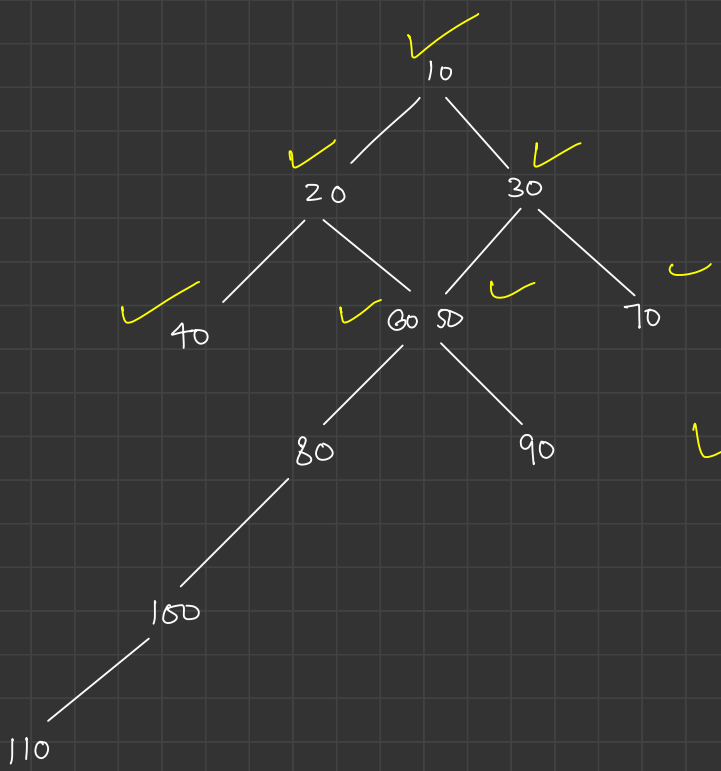
return this.node.data - o.node.data; → inc order

return this.vl - o.vl; → inc order



PQ → (Pair)

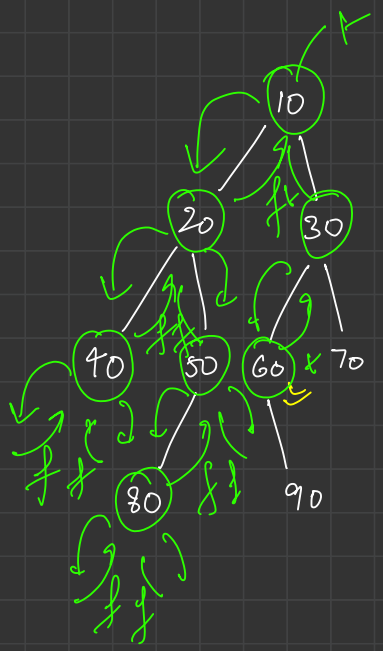
0 1 2 3 4 5



(40,1) (50,3) (60,3) (70,5)

(20,2) (30,4) (90,4) (80,2)

find a given node in a tree



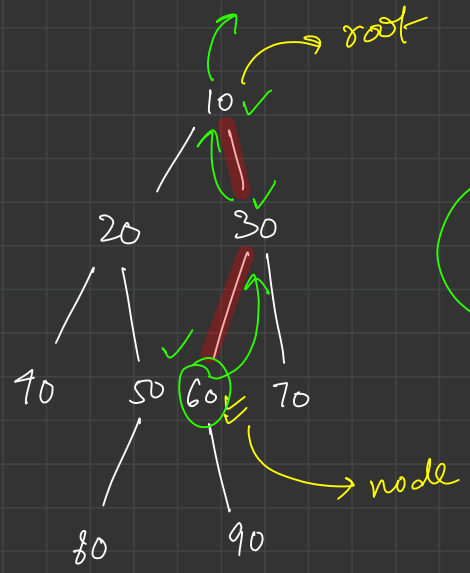

```
boolean find (Node root , int val)
{
    if (root == null)
        return false;

    if(root.data == val)
        return true;

    boolean filc = find(root.left, val);
    if(filc)
        return true;

    boolean firc = find(root.right, val);
    if(firc)
        return true;
}
return false;
```

Node to Root Path

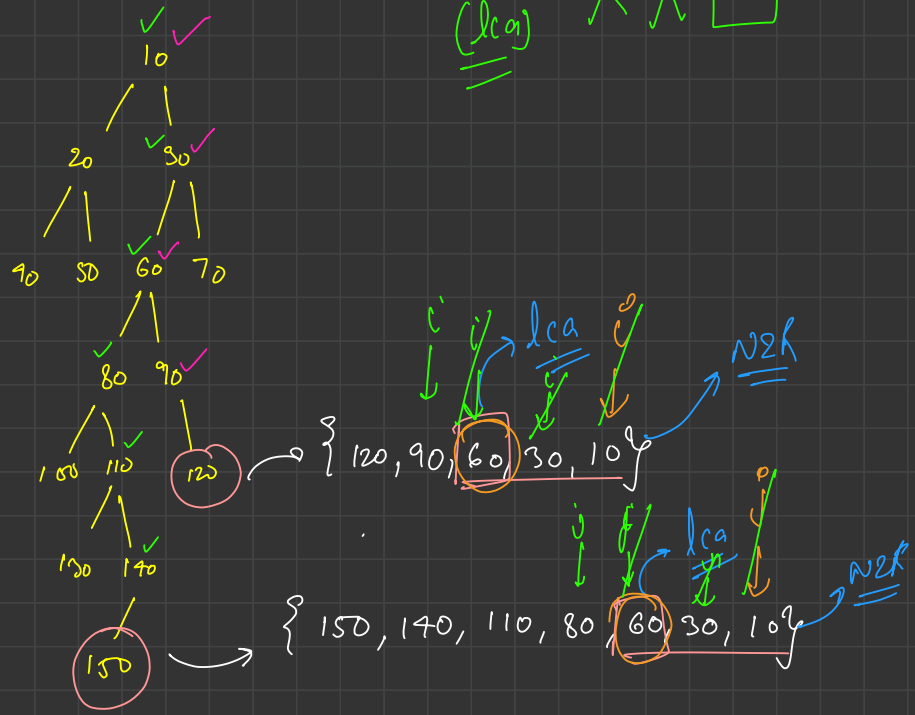


N2R

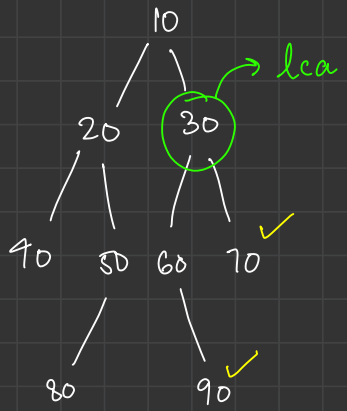
{60, 30, 10}

final fn

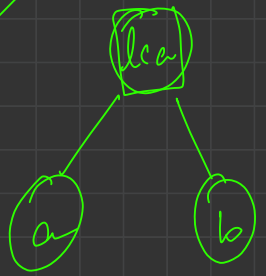
LCA (Lowest Common Ancestor)



Space Optimised

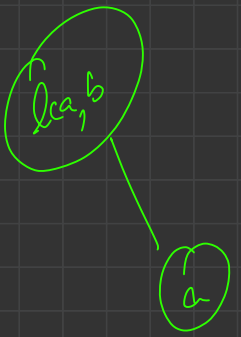


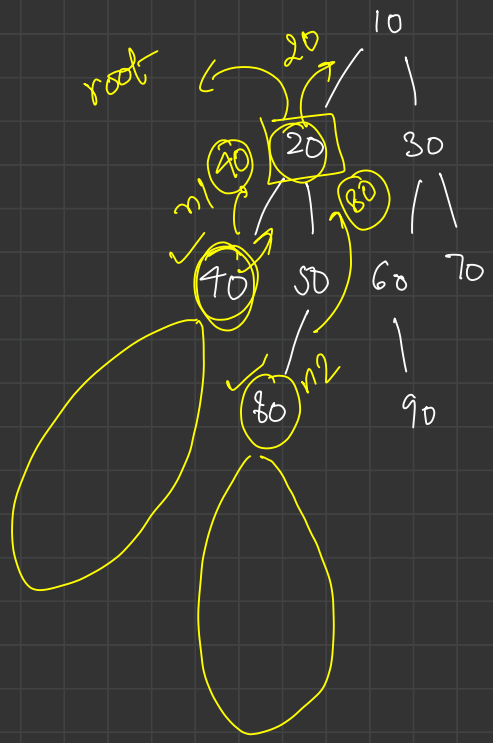
Case1



Case2

Case3



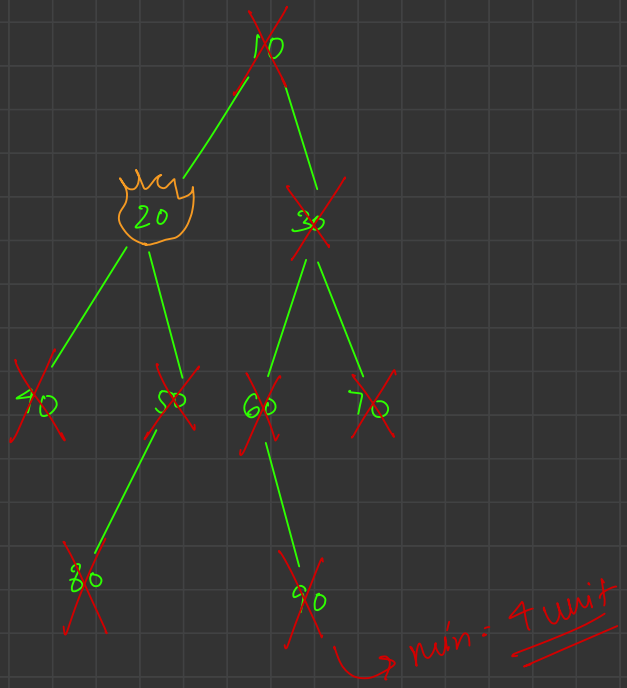


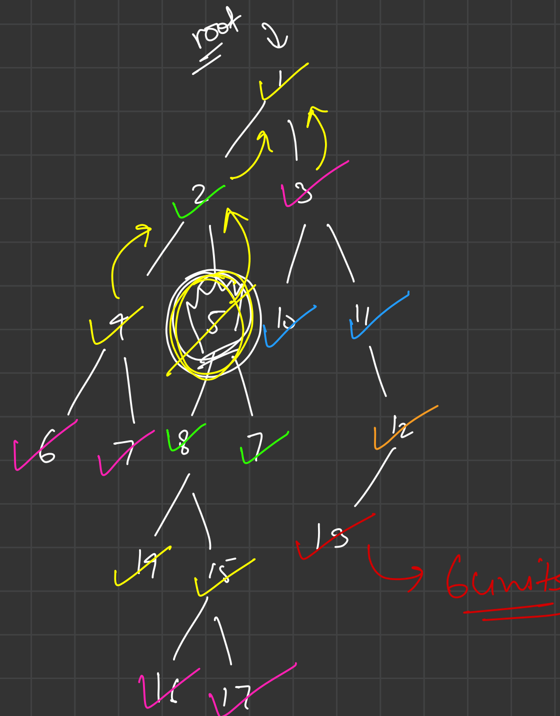
findLCA (n1, n2)

↓

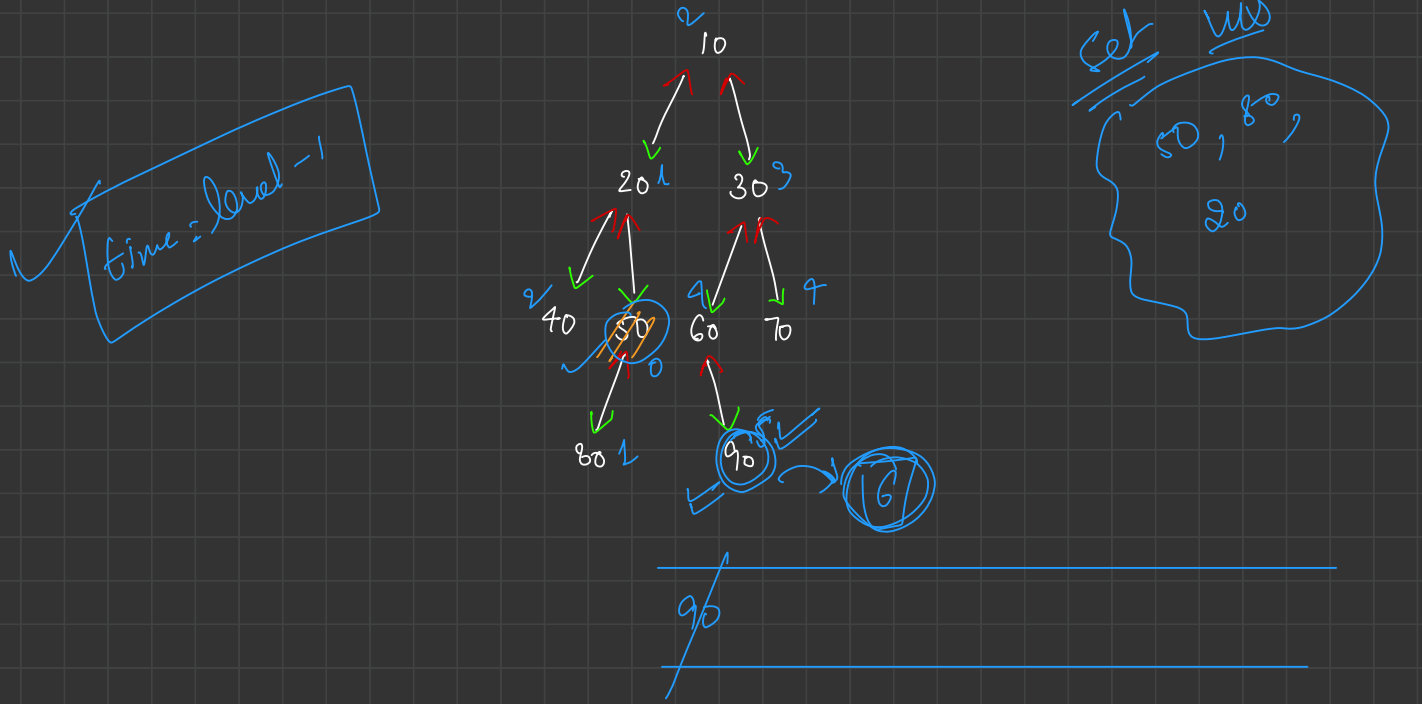
lca

Time to burn tree.





HashMap
↳ Key → child
value → parent


time : level - 1
2
10
20 1
30 3
2
40
50
0
4
60
70
4
80 1
90
16
set
list
50, 80,
20
90