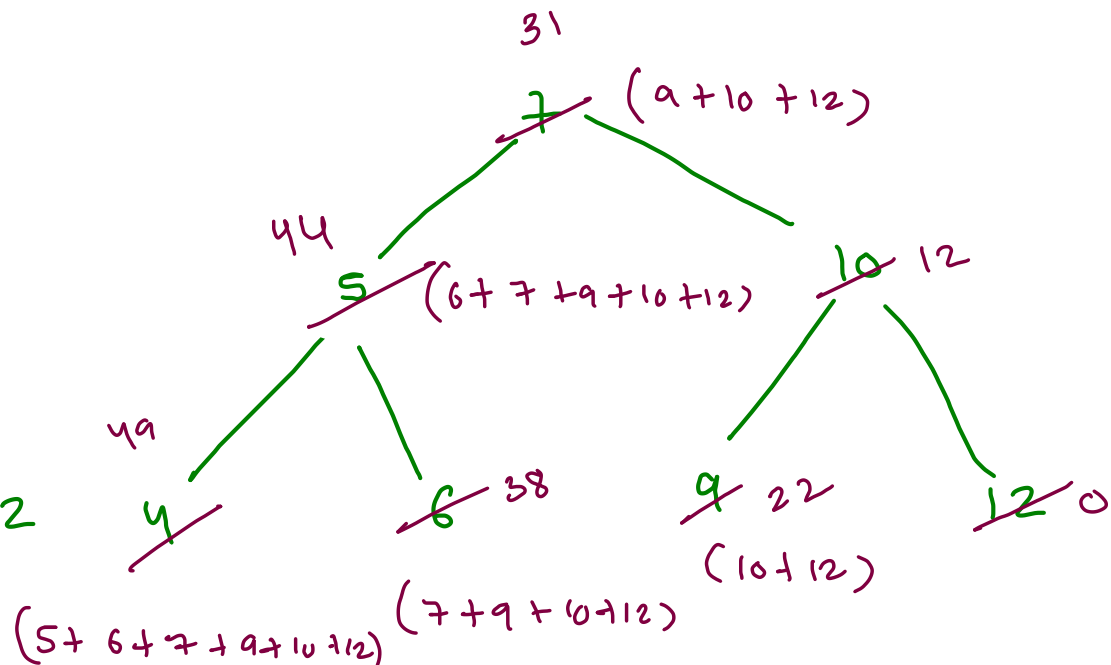
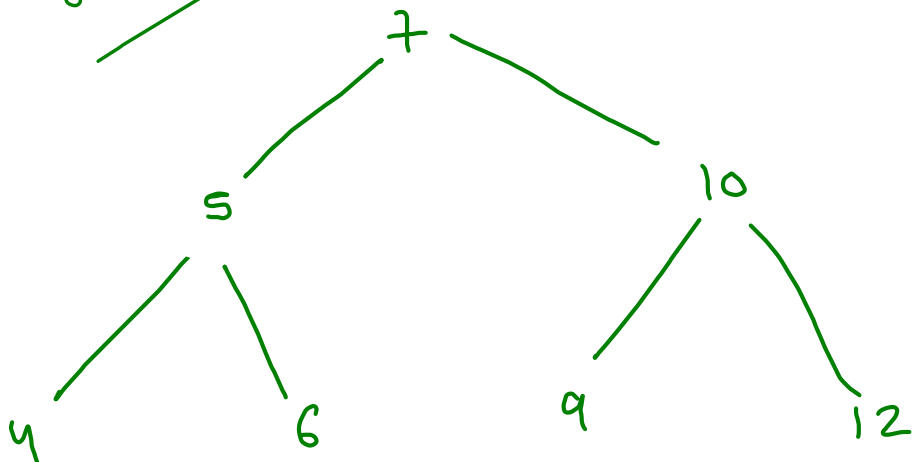


replace with  
sum of larger

left node right  $\rightarrow$  inorder



$$\text{Sum} = 0 + 12 + 10 + 9 + 7 + 6 + 5 + 4$$

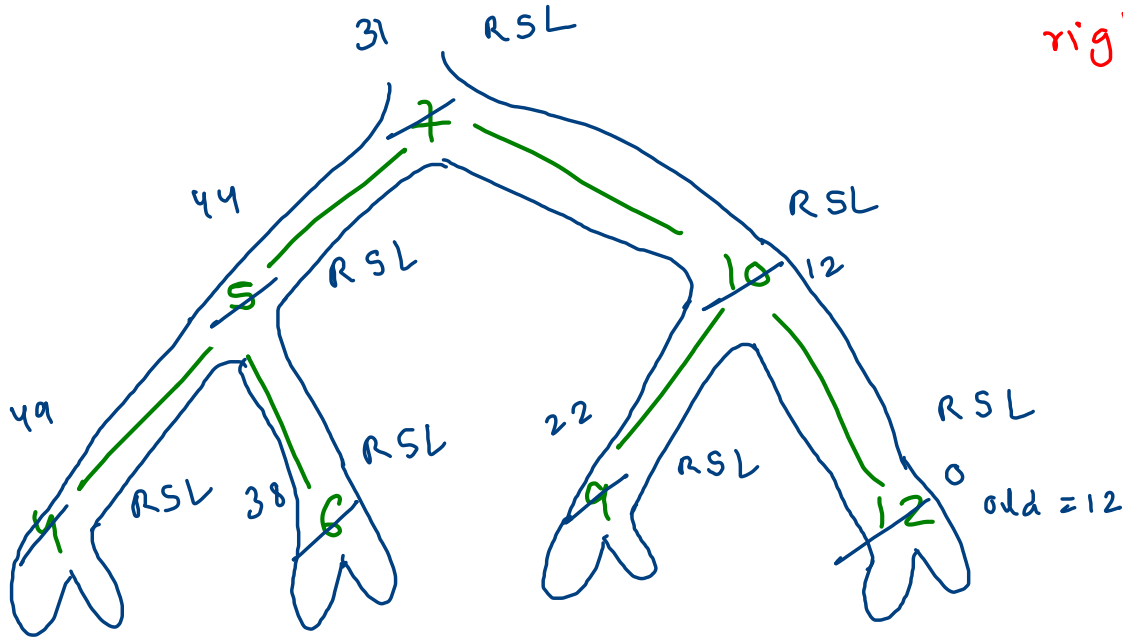
reverse

inorder  $\rightarrow$  dec order.

right

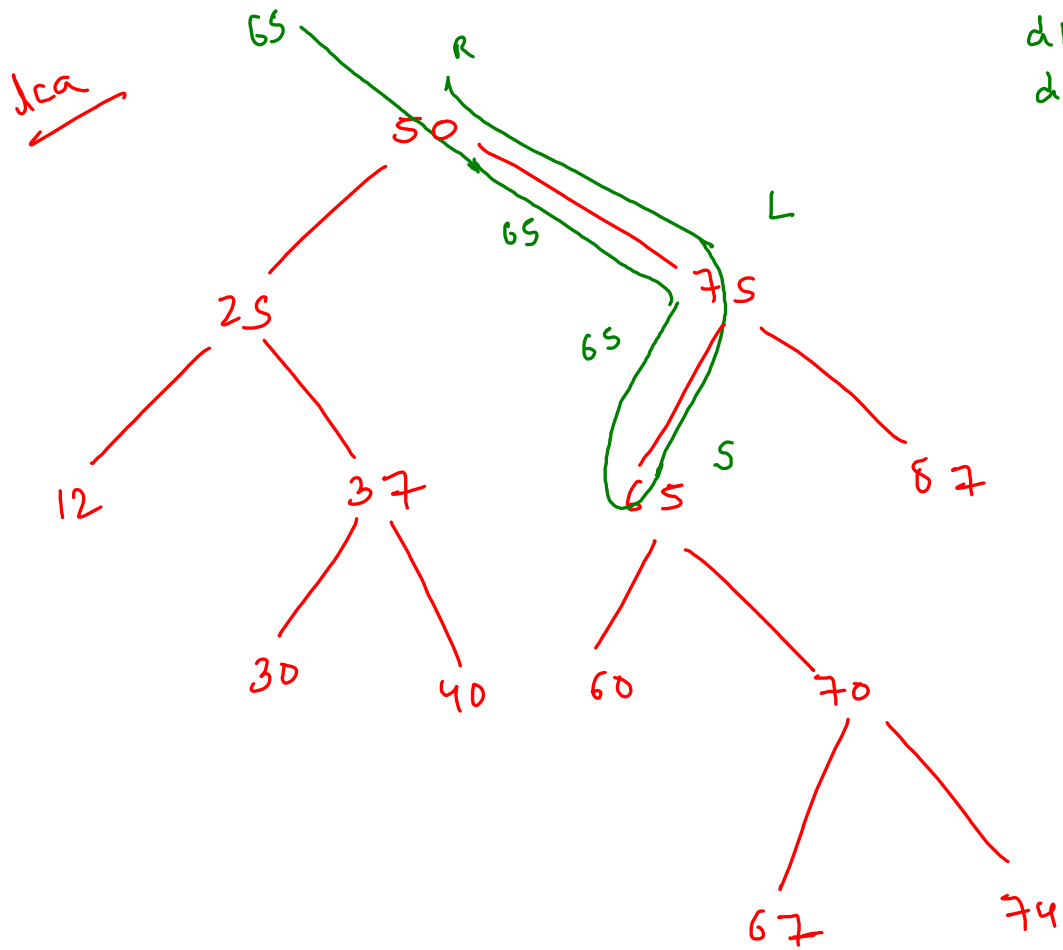
node

left



12 10 9 7 6 5 4

```
static int sum = 0;
public static void rwsol(Node node){
    if(node == null) {
        return;
    }
    R rwsol(node.right);
    S { int old = node.data;
      node.data = sum;
      sum += old;
    }
    L rwsol(node.left);
}
```

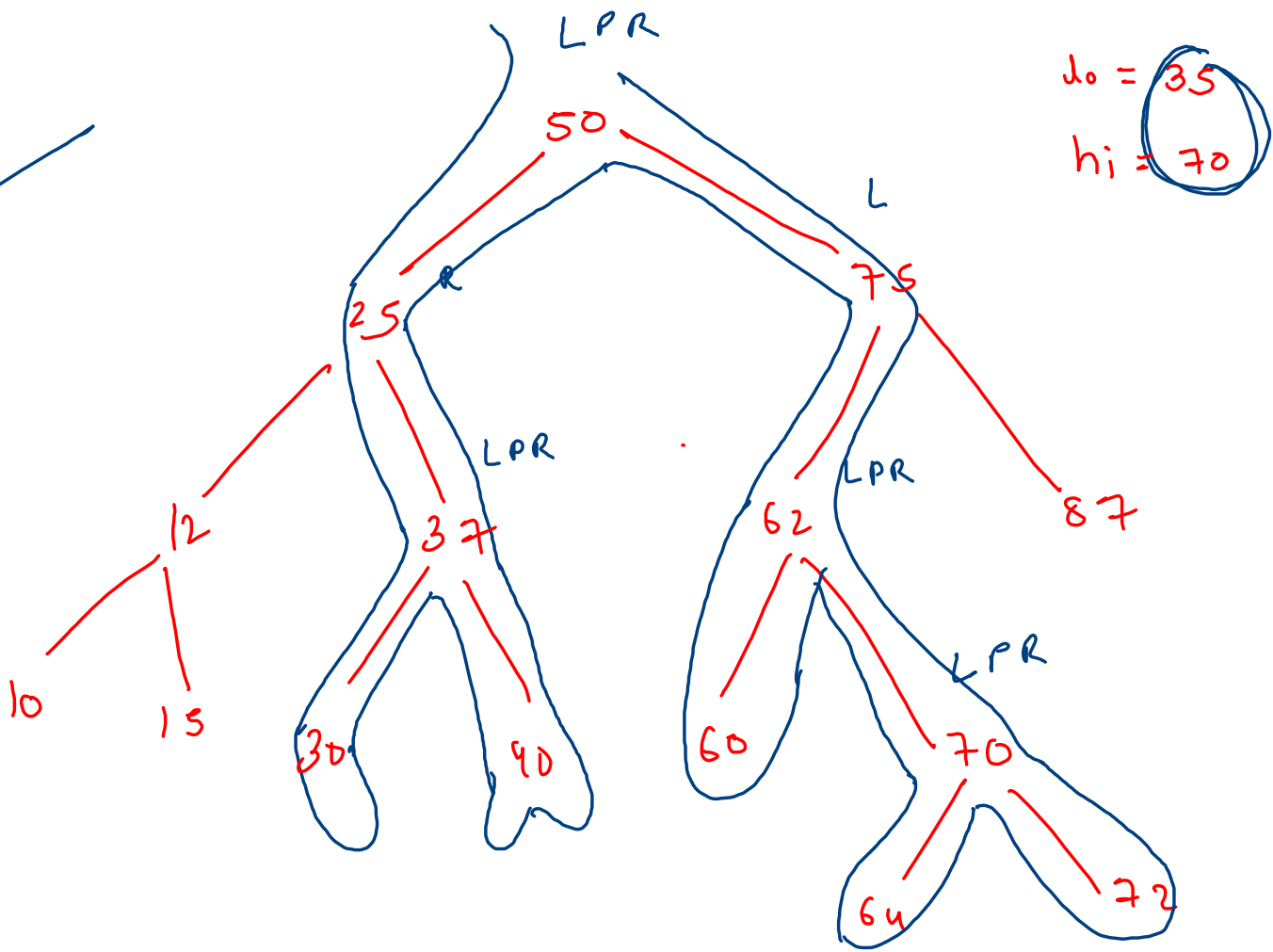


$\text{if } (d1, d2 > n.d) \{$   
 $\quad \text{return lca}(n.\text{right}, d1, d2)$   
 $\}$

$\text{else if } (d1, d2 < n.d) \{$   
 $\quad \text{return lca}(n.\text{left}, d1, d2)$   
 $\}$

$\text{else } \{$   
 $\quad \text{return node.data}$   
 $\}$

print in  
range



longest sum  
pair

time space

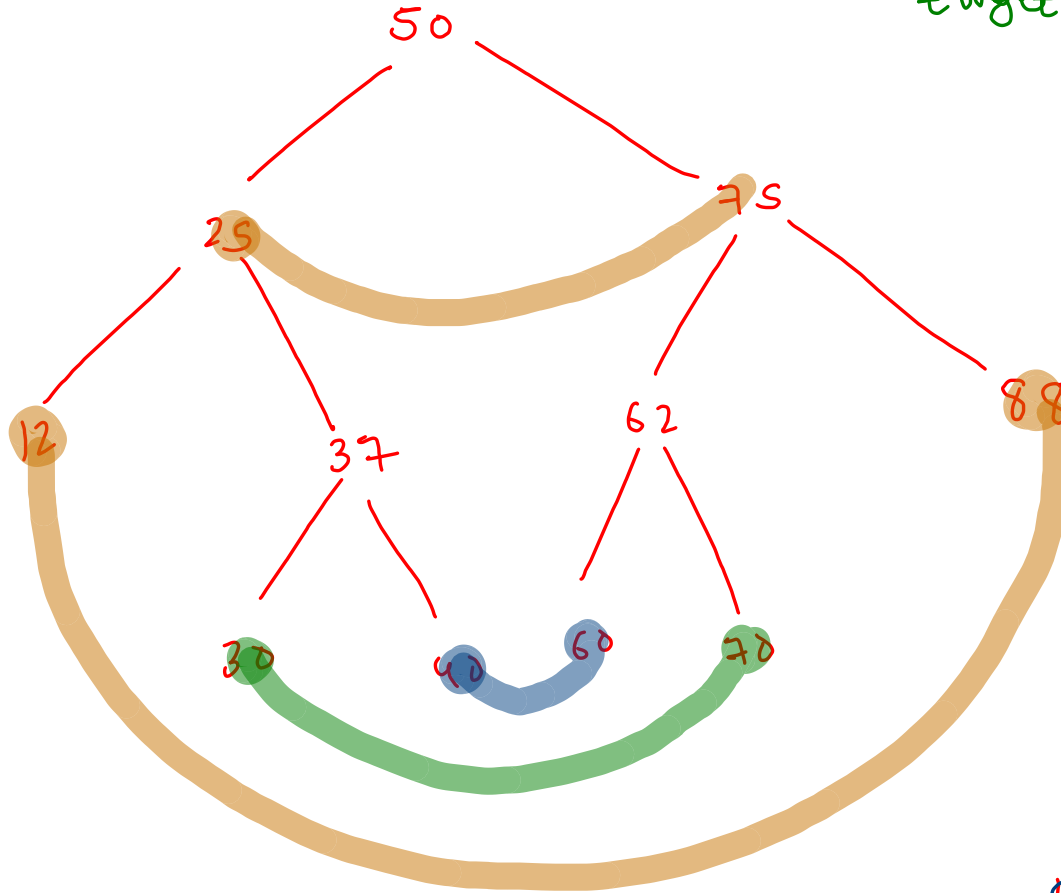
(i)  $n \log n$   $\log n$

(ii)  $n$   $n$

(iii)  $n$   $\log n$

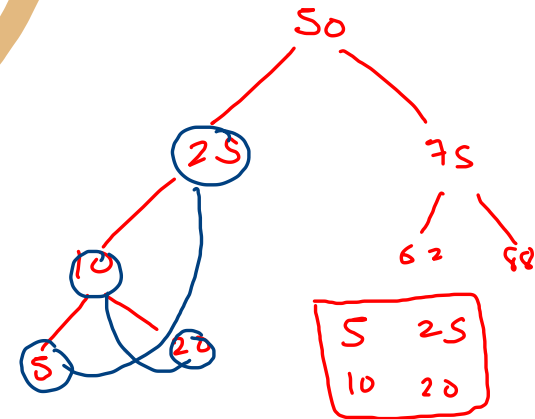
( $\log n = \text{height}$ )

target = 100

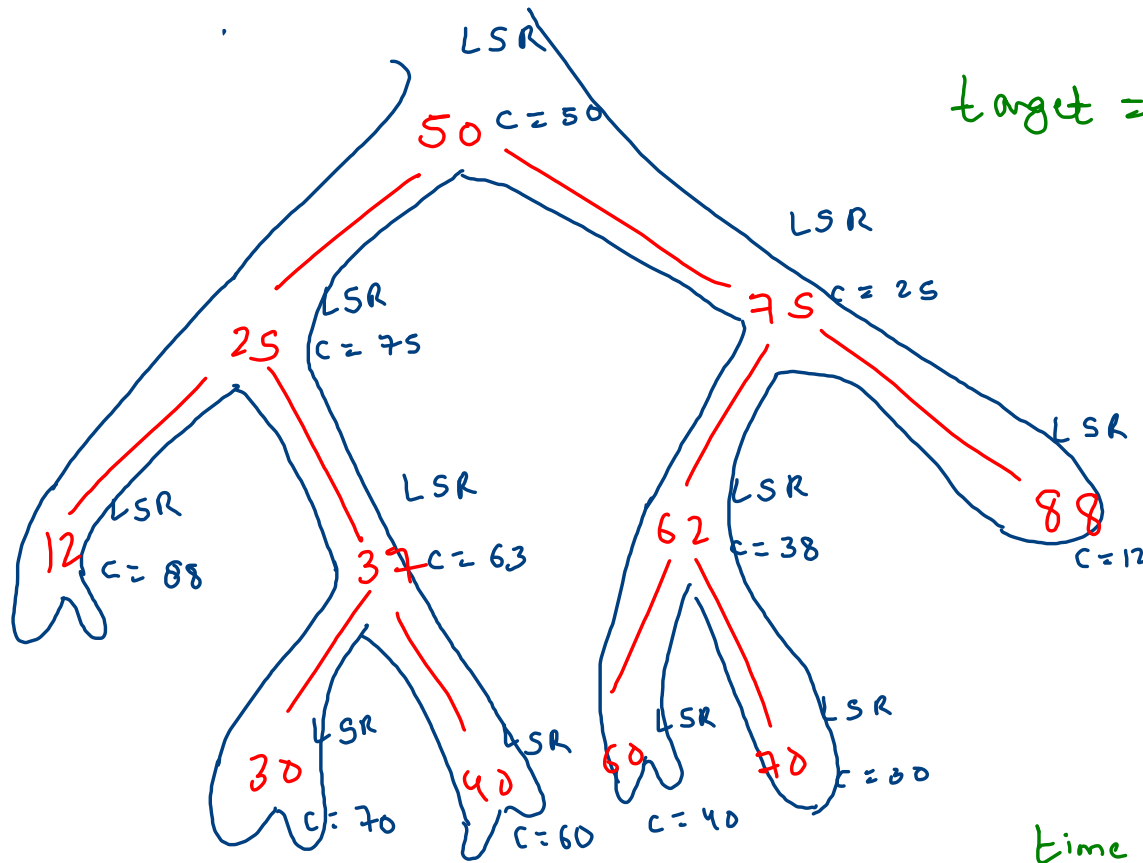


12	8
25	75
30	70
40	60

tan = 30



5	25
10	20



target = 100 , oroot = 50k

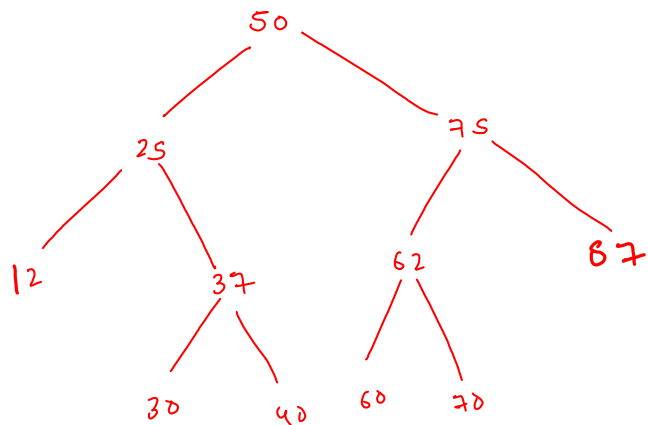
```

public static void targetSumPair1(Node node, int target, Node oroot) {
    if (node == null) {
        return;
    }
    L [ targetSumPair1(node.left, target, oroot);
    S [ int comp = target - node.data;
        if (comp > node.data) {
            boolean fit = find(oroot, comp);
            if (fit == true) {
                System.out.println(node.data + " " + comp);
            }
        }
    R [ targetSumPair1(node.right, target, oroot);
}
  
```

Time =  $n \log n$

Space =  $\log n$

12	88
25	75
30	70
40	60



space  $\rightarrow O(n)$

time  $\rightarrow O(n)$

target = 100

[12, 25, 30, 37, 40, 50, 60, 62, 70, 75, 87]

if (dist[do] + dist[hi] > target)?

hi--;

} else if (dist[do] + dist[hi] < target)?  
do++;

} else {  
print(pair); do++; hi--;

}

↑

do

↑

hi

25 75

30 70

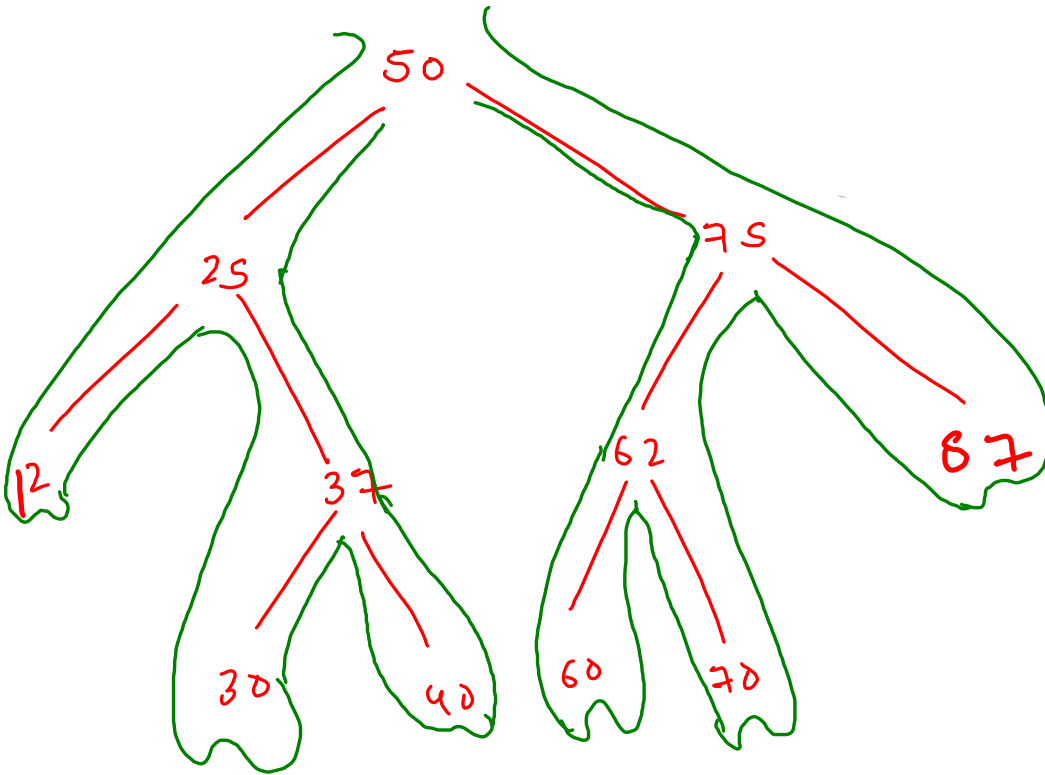
40 40

time  $\rightarrow O(n)$

space  $\rightarrow \log n$

left = 50

right = 50



target = 100

25

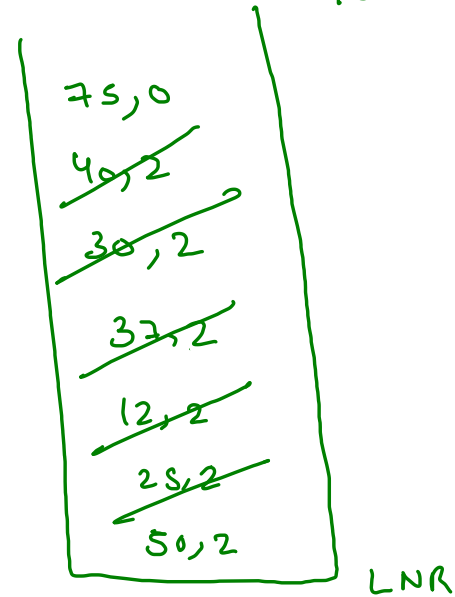
75

30

70

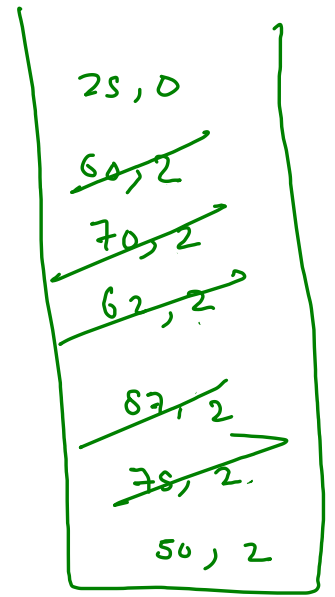
40

60



LNR

LS (regular  
inorder)



RS RNL  
(reverse  
inorder)



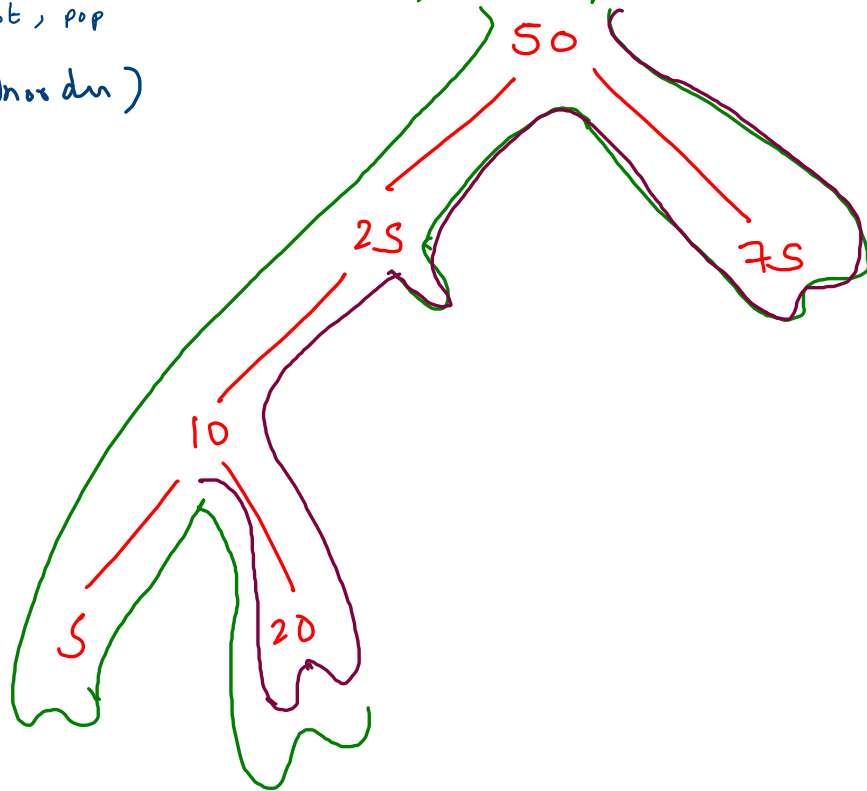
0  $\rightarrow$  pre, left child, st++

1  $\rightarrow$  in, right child, st++

2  $\rightarrow$  post, pop

(Inorder)

0  $\rightarrow$  pre, right child, st++  
1  $\rightarrow$  in, left child, st++  
2  $\rightarrow$  post, pop.



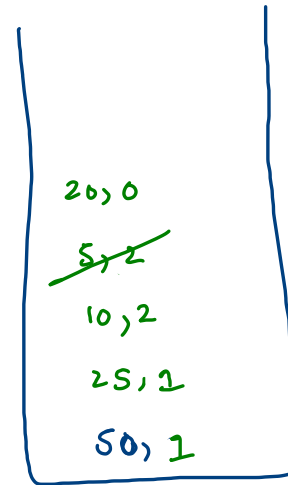
target = 30

left = 20

right = 10

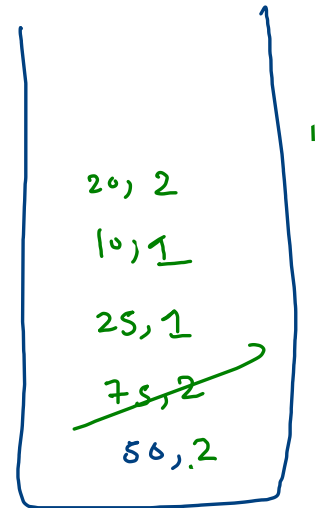
5 25

10 20



ds (LNR)

inorder



rs (RNL)

reverse inorder