

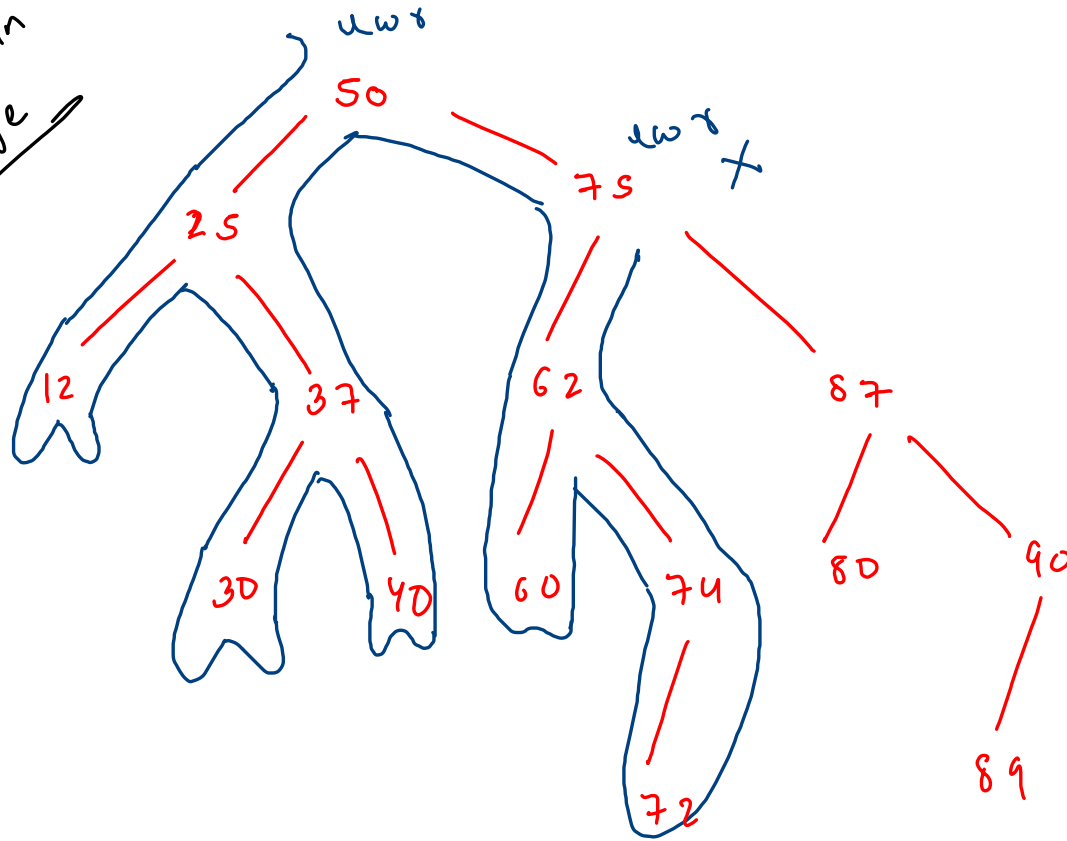
LCA

```
public static int lca(Node node, int d1, int d2) {  
    if(d1 > node.data && d2 > node.data) {  
        return lca(node.right, d1, d2);  
    }  
    else if(d1 < node.data && d2 < node.data) {  
        return lca(node.left, d1, d2);  
    }  
    else {  
        return node.data;  
    }  
}
```

d1 =

d2 =

Print in  
range



$l_0 = 30$

$h_i = 65$

call (node-left);

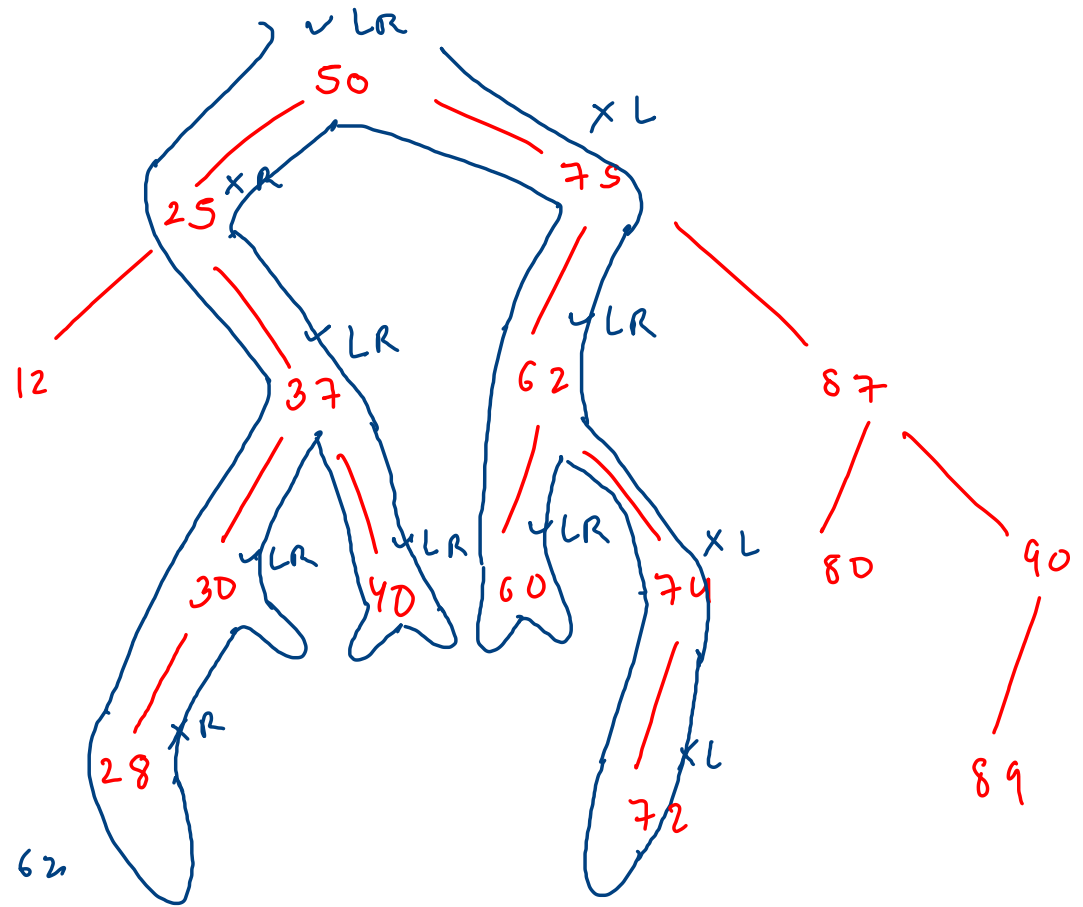
if ( $l_0 \leq \text{node.data} \leq h_i$ )  
 syso (node.data);

}

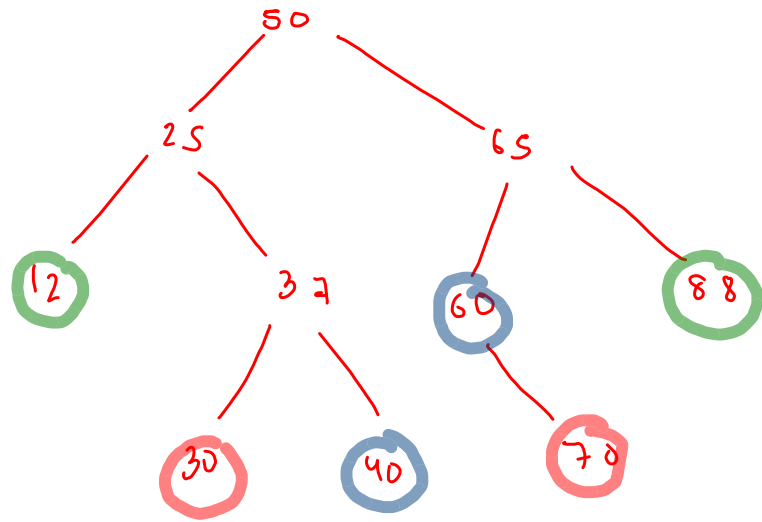
call (node-right);

lo = 30 hi = 65

```
public static void pir(Node node, int lo, int hi) {  
    if(node == null) {  
        return;  
    }  
  
    if(node.data >= lo && node.data <= hi) {  
        pir(node.left, lo, hi);  
        System.out.println(node.data);  
        pir(node.right, lo, hi);  
    }  
    else if(node.data < lo) {  
        pir(node.right, lo, hi);  
    }  
    else if(node.data > hi) {  
        pir(node.left, lo, hi);  
    }  
}
```

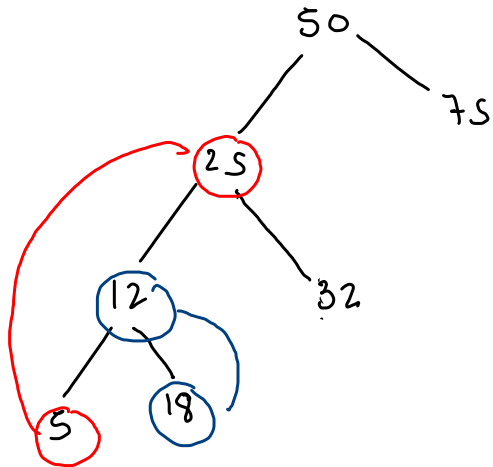


30 37 40 50 60 62



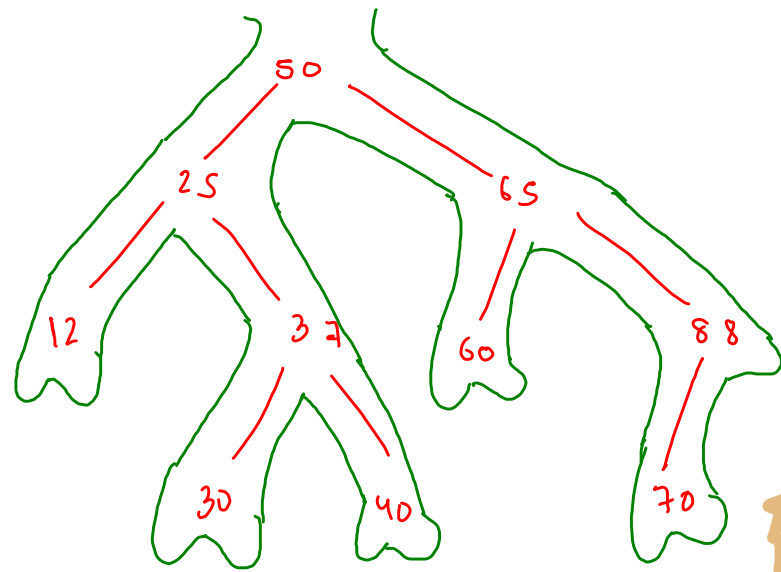
target = 100

12	88
30	70
40	60



target = 30

5	25
12	18



target = 100

call (node.left);

// work

int rem\_tar = tar - node->data;

if (rem\_tar > node->data) { // to avoid duplicate pairs

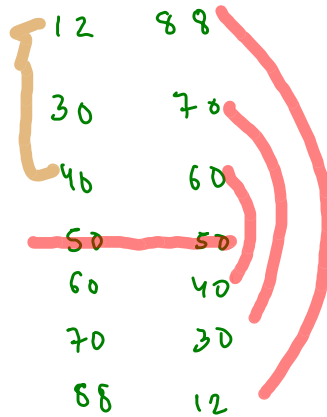
boolean ans = find (root, rem\_tar);

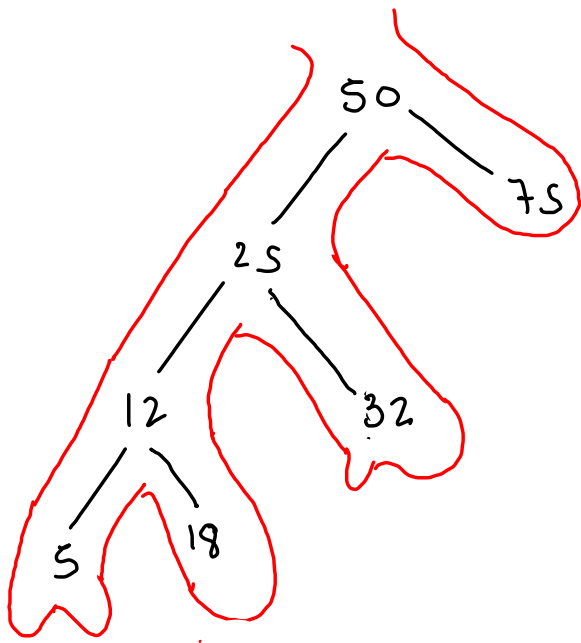
if (ans == true)

syso (node->data , rem\_tar);

}

call (node->right);





5      25  
12     18

call (node.left);

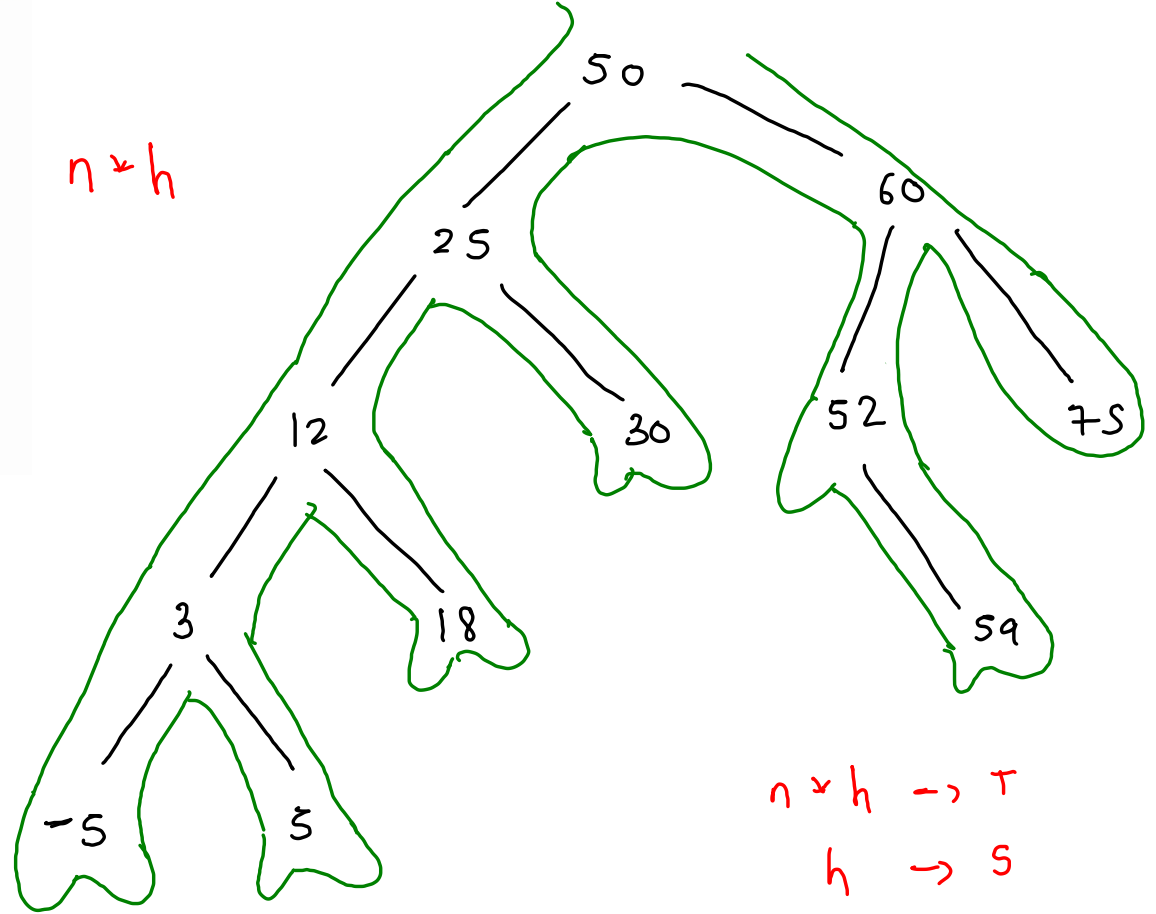
// work

```
int rem_tan = tan - node.data;  
if (rem_tan > node.data) {  
    boolean ans = find (root, rem_tan);  
    if (ans == true)  
        syso (node.data, rem_tan);  
}
```

call (node.right);

target = 55

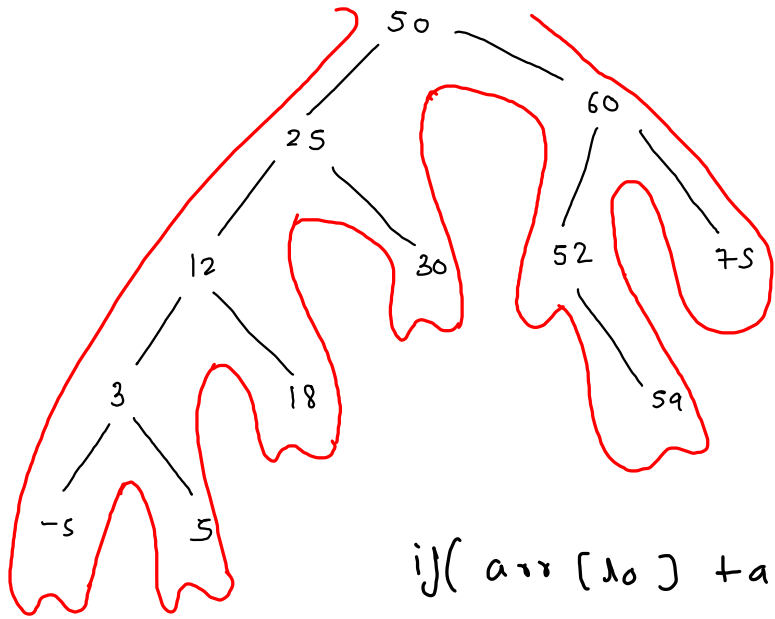
$n \times h$



$n \times h \rightarrow T$   
 $h \rightarrow S$

```
public static void targetSumPair(Node node, Node root, int target) {  
    if (node == null) {  
        return;  
    }  
  
    targetSumPair(node.left, root, target);  
  
    //work  
    int rem_tar = target - node.data;  
  
    if (rem_tar > node.data) {  
        if (find(root, rem_tar) == true) {  
            System.out.println(node.data + " " + rem_tar);  
        }  
    }  
  
    targetSumPair(node.right, root, target);  
}
```

-5 60  
3 52  
5 50  
25 30



$T: O(n)$

$S: O(n)$

target = 55

-5 3 5 12 18 25 30 50 52 59 60 75  
 hi lo

if( arr[lo] + arr[hi] == target )

syso( arr[lo] + " " + arr[hi] );  
 lo++, hi--;

else if( arr[lo] + arr[hi] > target ) {  
 hi--;

}  
 else {  
 lo++;

-5 60  
 3 52  
 5 50  
 25 30