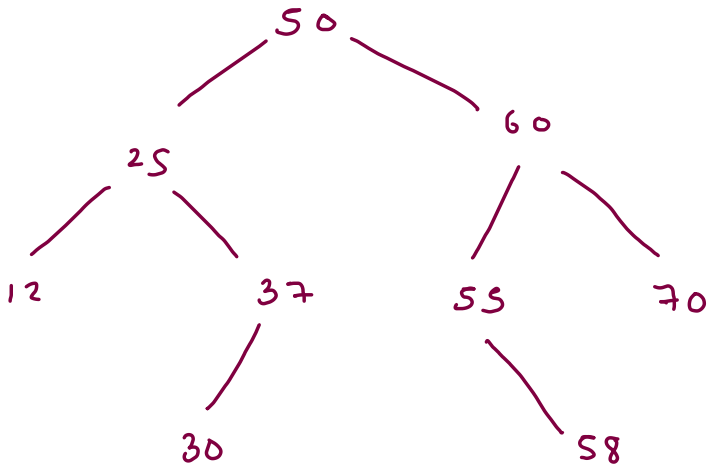
BST



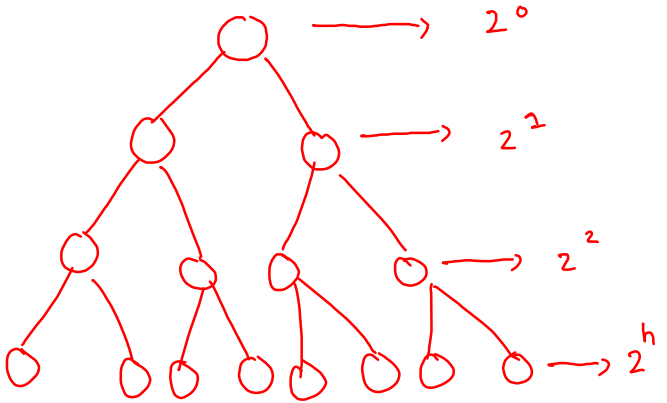
find , lca → $\Theta(h)$

advantage $h \approx \log n$

proof $h \approx \log n$

$n \rightarrow$ no. of nodes

$h \rightarrow$ height of tree



$n = 2^0 + 2^1 + 2^2 + \dots + 2^h$

$t = h+1$

$r = 2$

$a = 1$

$n = 2^{h+1} - 1$

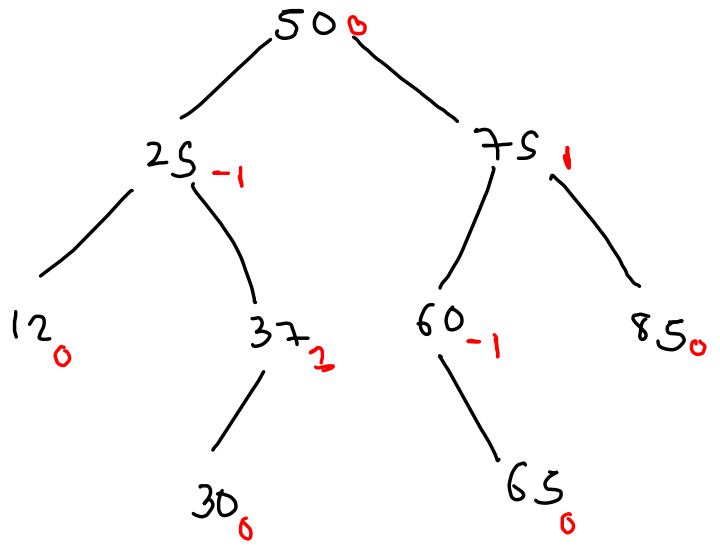
$n = 2^{h+1}$

$\frac{a(r^t - 1)}{r-1} = \frac{1(2^{h+1} - 1)}{1}$

$\log_2 n = h+1$

$h \propto \log_2 n$

AVL : balanced BST



$bf \leq |1|$

bf : $lh - rh$

ht : edges

Safe bf : 0, -1, 1

```java
public static Node add(Node node,int data) {
    if(node == null) {
        return new Node(data);
    }

    if(node.data < data) {
        node.right = add(node.right,data);
    }
    else if(node.data > data) {
        node.left = add(node.left,data);
    }
    else {
        //do nothing
    }

    updateHeightAndBalance(node);
    return node;
}
```

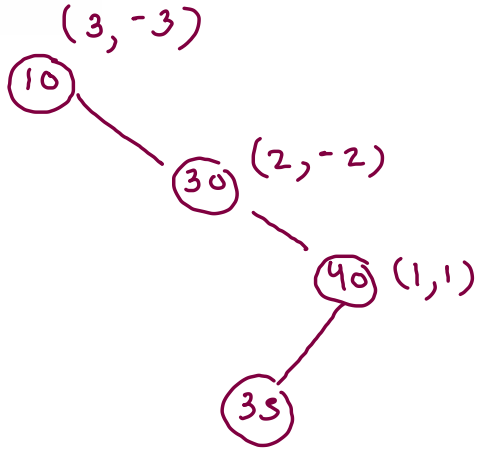
```java
public static void updateHeightAndBalance(Node node) {
    if(node == null) {
        return;
    }

    int lht = node.left == null ? -1 : node.left.ht;
    int rht = node.right == null ? -1 : node.right.ht;

    int ht = Math.max(lht,rht) + 1;
    int bf = lht - rht;

    node.ht = ht;
    node.bf = bf;
}
```

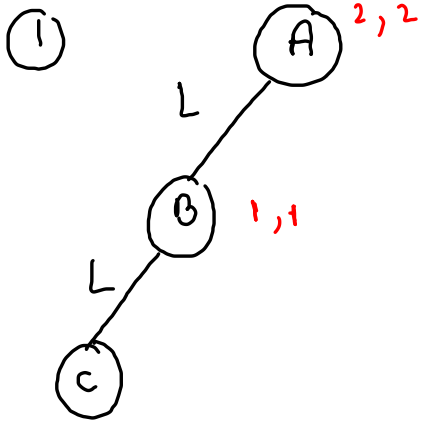
ht, bf

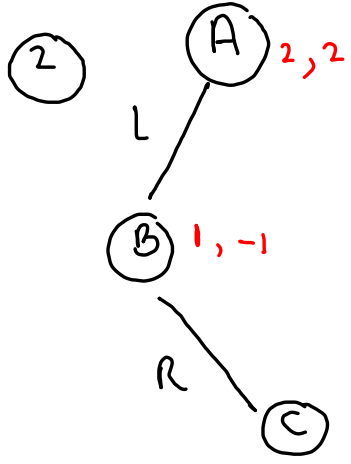


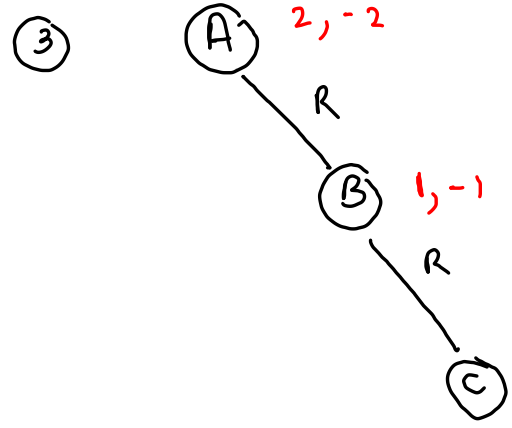

```
D:\Workspace\Rep
5 <- 10 -> 30
. <- 5 -> 8
. <- 8 -> .
. <- 30 -> 40
35 <- 40 -> 90
. <- 35 -> .
50 <- 90 -> .
. <- 50 -> .
```

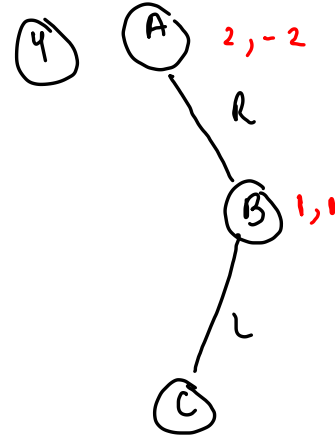
```java
root = add(root,10);
add(root,30);
add(root,40);
add(root,35);
add(root,90);
add(root,5);
add(root,50);
add(root,8);
```

mistakes (the first node where bf > 1 1)





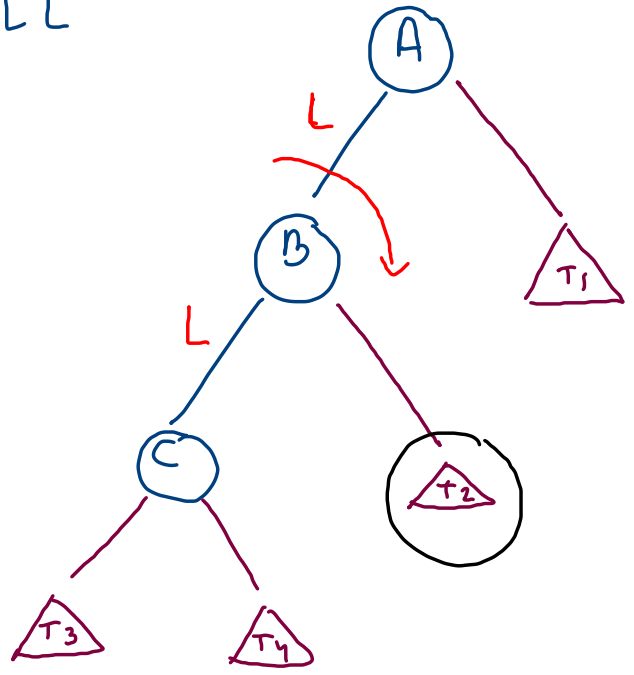




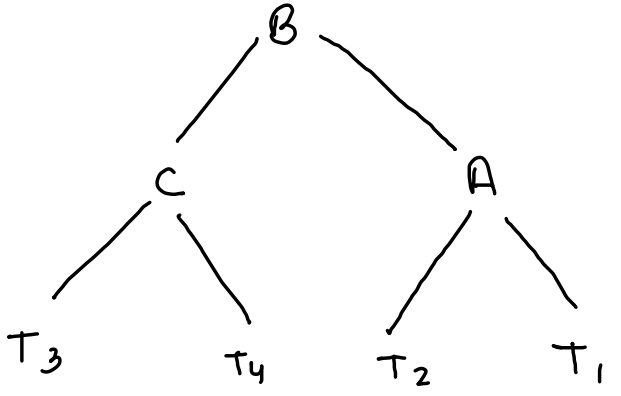
(ht, bal)

Solve the problems:

① LL

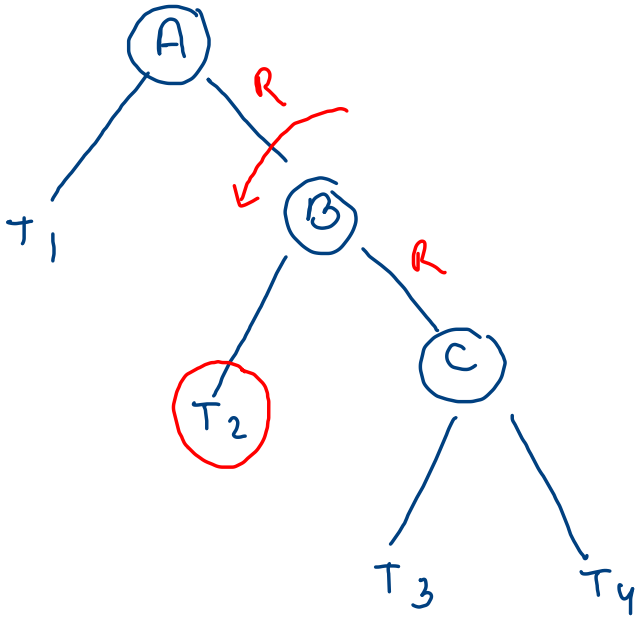


right rotation at B →

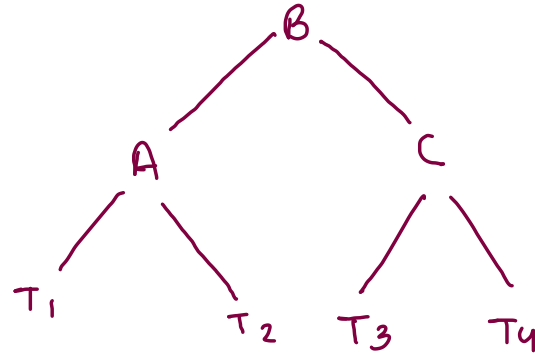



```
Node temp = B.right;
B.right = A;
A.left = temp;
return B;
```

(2). RR

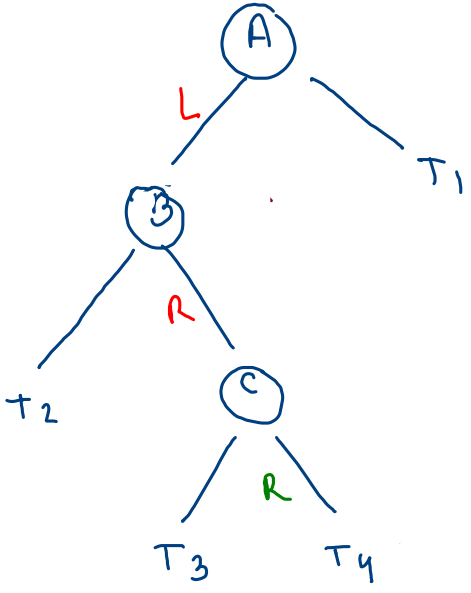


left rotation at B



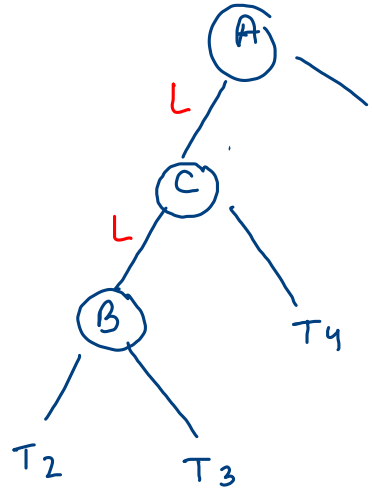

```
Node temp = B.left;
B.left = A;
A.right = temp;
return B;
```

③ LR



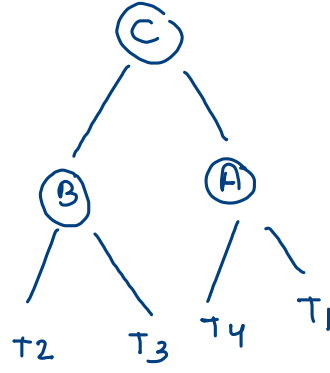
Solve RR problem B

left rotation at C

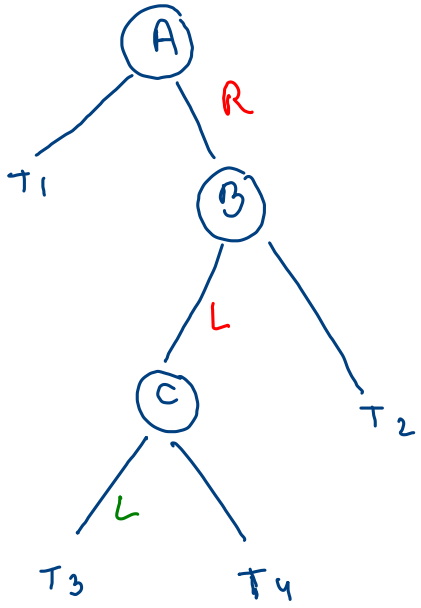


solve LL at A

right rotation at C

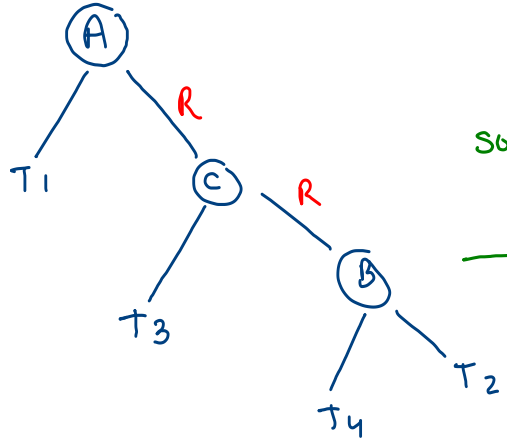


④ RL



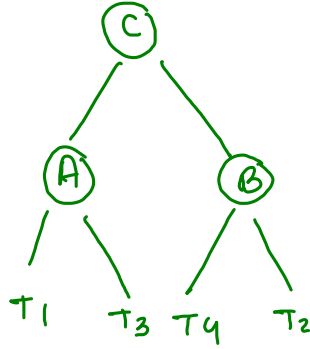
solve LL at B

right rotation at C



Solve RR at A

left rotation at C



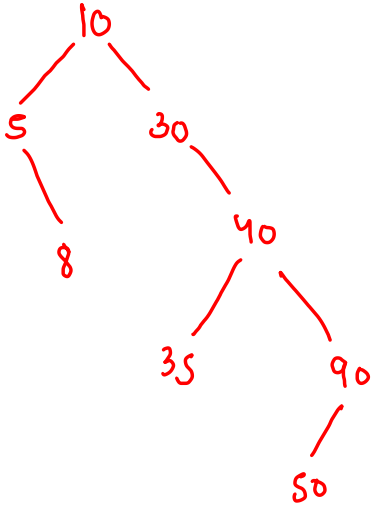
```java
public static void main(String[]args) {
    Node root = null;
    int[]arr = {10,30,40,35,90,5,50,8};

    for(int i=0; i < arr.length;i++) {
        root = add(root,arr[i]);
    }

    display(root);
}
```

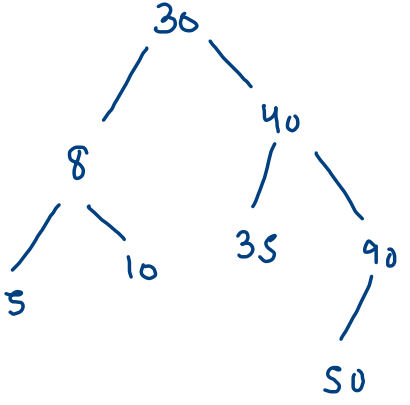
```
5 <- 10 -> 30
. <- 5 -> 8
. <- 8 -> .
. <- 30 -> 40
35 <- 40 -> 90
. <- 35 -> .
50 <- 90 -> .
. <- 50 -> .
```

before (AVL)

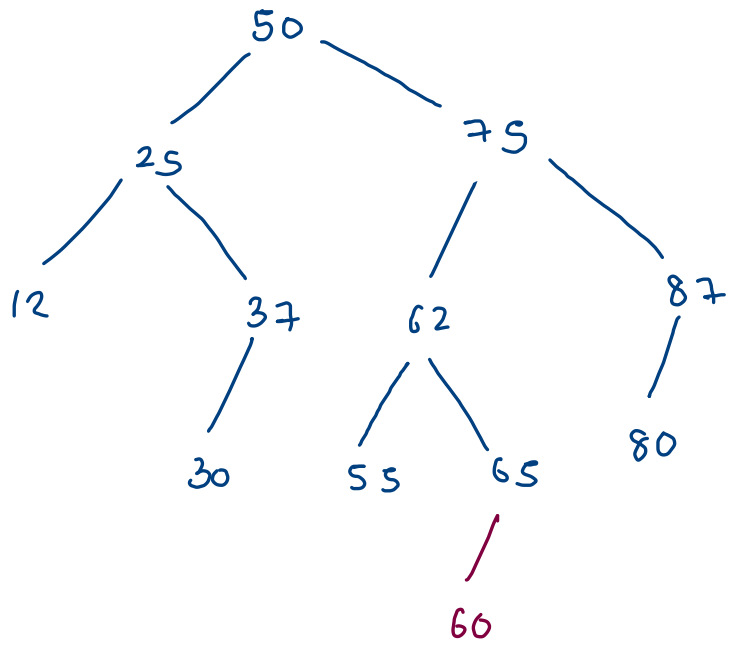



```
8 <- 30 -> 40
5 <- 8 -> 10
. <- 5 -> .
. <- 10 -> .
35 <- 40 -> 90
. <- 35 -> .
50 <- 90 -> .
. <- 50 -> .
```

after (AVL)



deletion



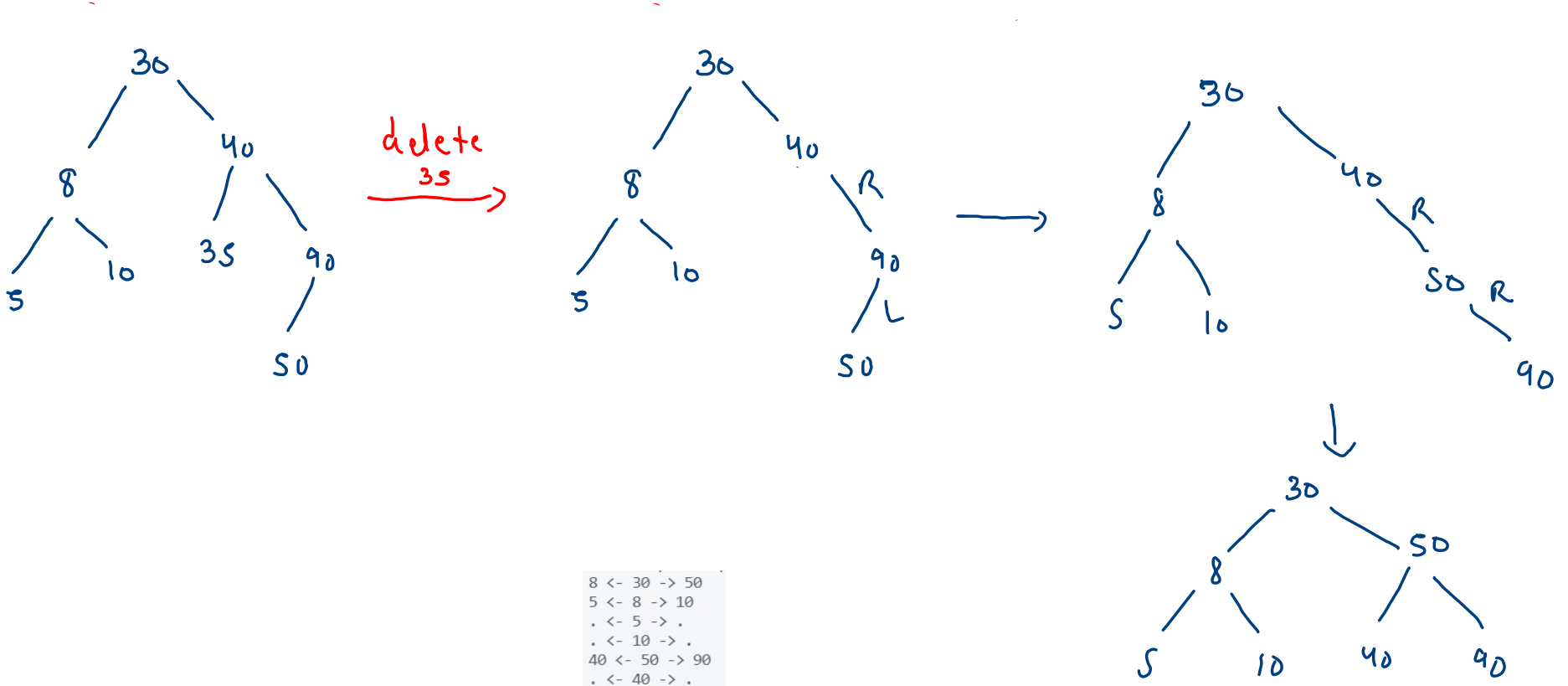
no child : return null

single child : return single child

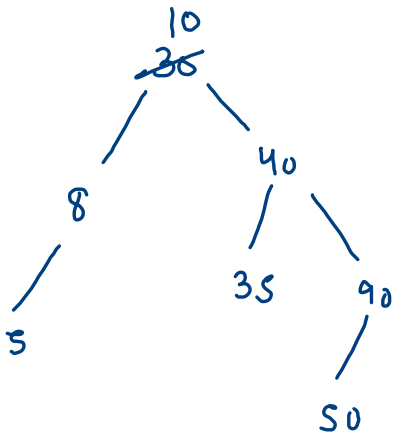
both child :

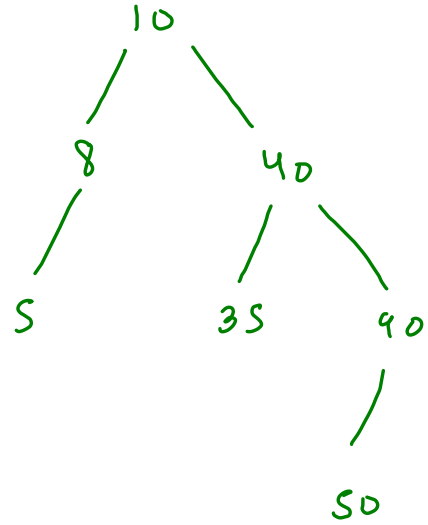
(i) replace node.data with left max.

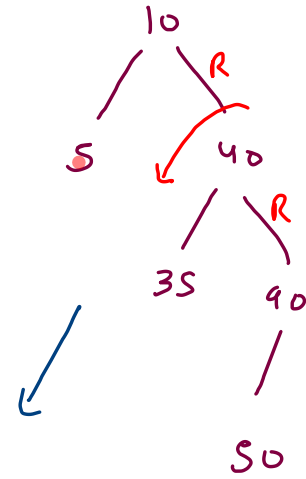
(ii) deletion(node.left, max)




```
8 <- 30 -> 50
5 <- 8 -> 10
. <- 5 -> .
. <- 10 -> .
40 <- 50 -> 90
. <- 40 -> .
. <- 90 -> .
```








```
10 <- 40 -> 90
5 <- 10 -> 35
. <- 5 -> .
. <- 35 -> .
50 <- 90 -> .
. <- 50 -> .
```