

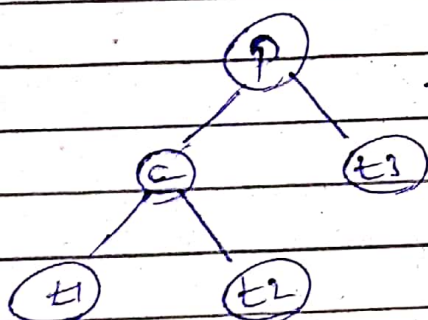
# AVL (Adelson-Velsky and Landis)

\* Algorithm Used for  
Self Balancing Binary Tree

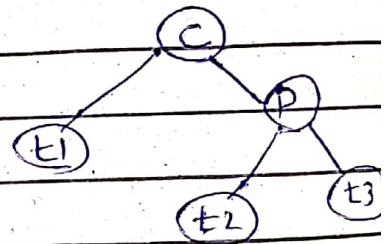
\* Two approaches or operations

Left Rotate  
Right Rotate

\* Example of operations



right rotate  
from P



Left rotate  
from C

\* 4 rules to perform operations

Left-left case  
Left-right case  
Right-left case  
Right-right case

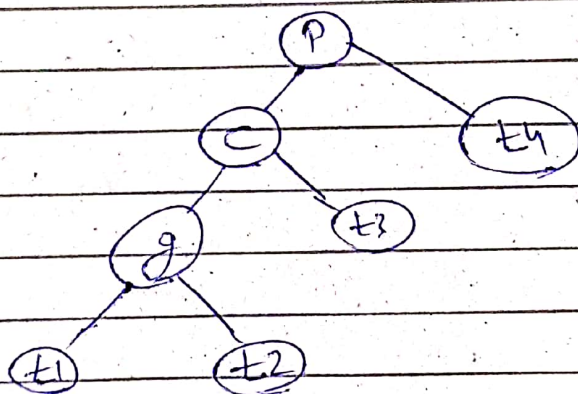
①

## Left - left Case

Child is at left side of parent.

Grandchild is at left side of child.

### Representation



### Terminologies

P → parent

C → child

g → grand child

t1 → SubTree 1

t2 → SubTree 2

t3 → SubTree 3

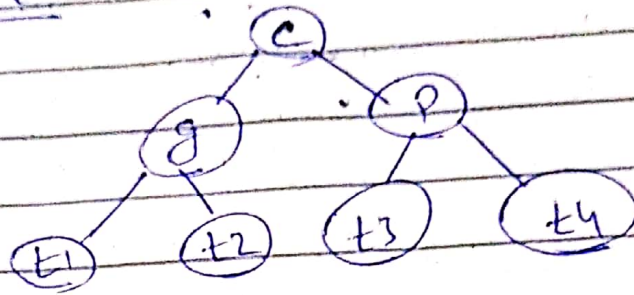
t4 → SubTree 4

### Operations Performed in this case

↳ Right Rotate (P)

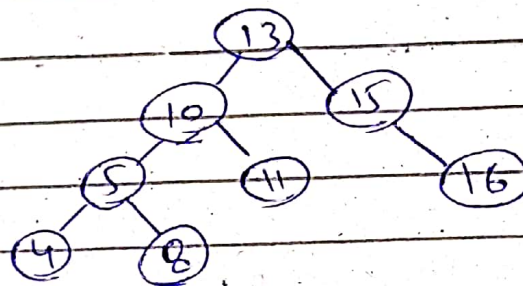


Result

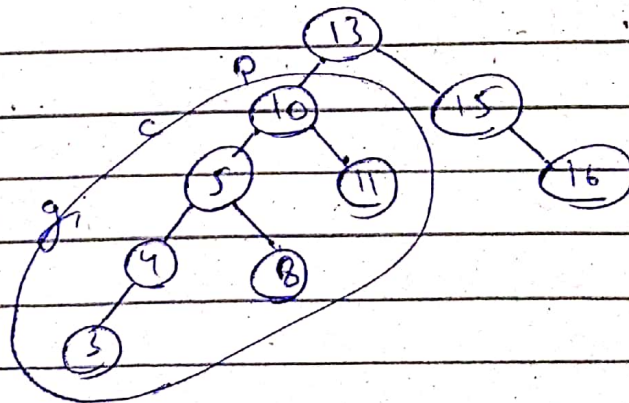


Example:-

Tree Given as

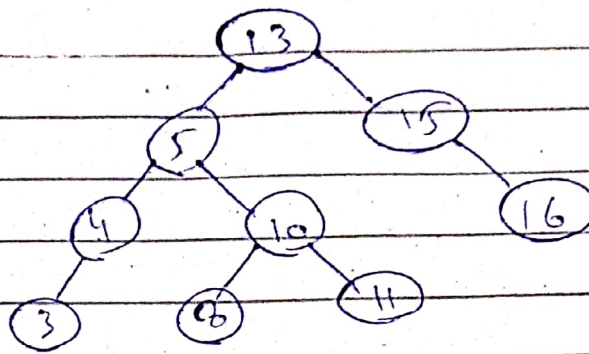


Adding 3 in this



The rounded subtree is unbalanced.  
Grandchild will be at the side  
where changes are made. Since 3  
is getting added on left side so  
grandchild will be at left

Right rotating (p)



It is now Balanced

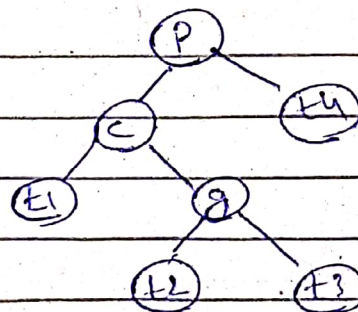
②

Left Right Case:

→ Child is at left side of parent

GrandChild is at right side of child

Representation



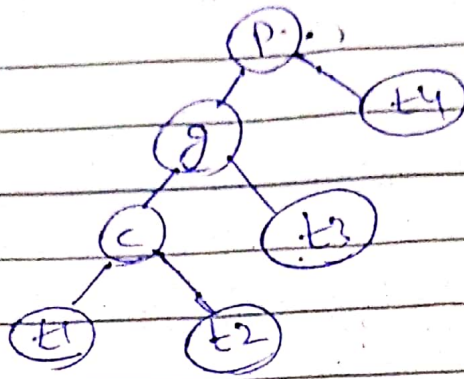
Operations Performed in this case

→ Left Rotate (C)

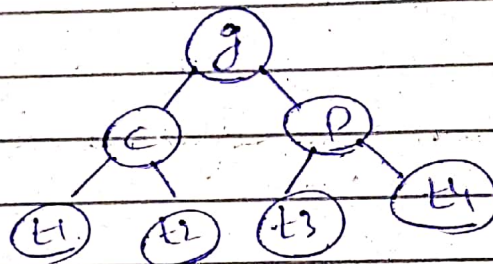
Right Rotate (P)

Result After Left rotating (C)

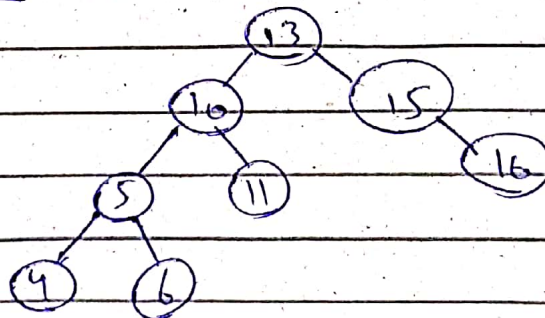




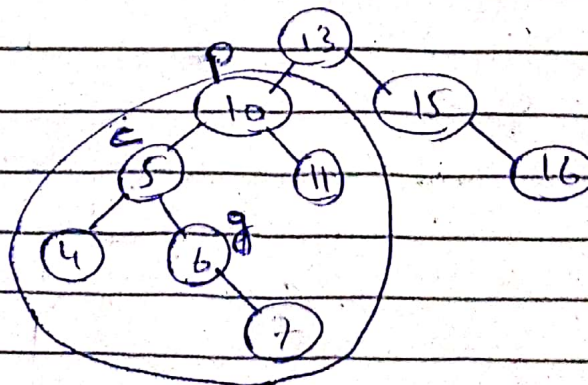
After Right rotating (P)



→ Example



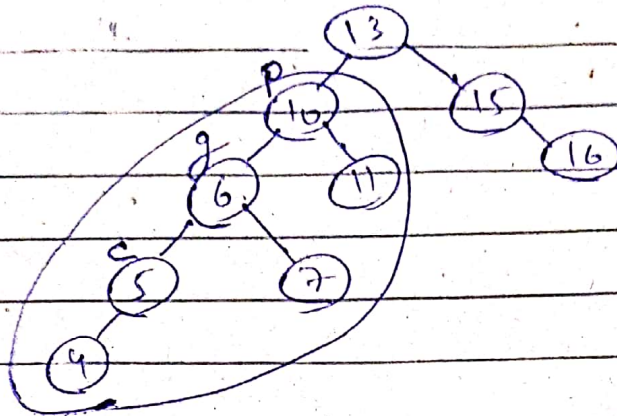
Adding 7 into it



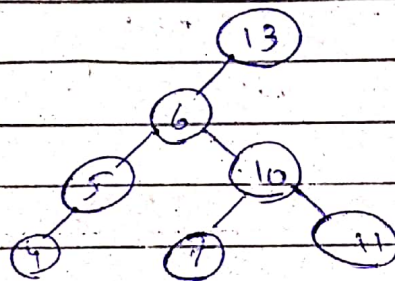
→ Applying operations

1 - Left - Rotating (C)

Note



2 - Right rotating (p)



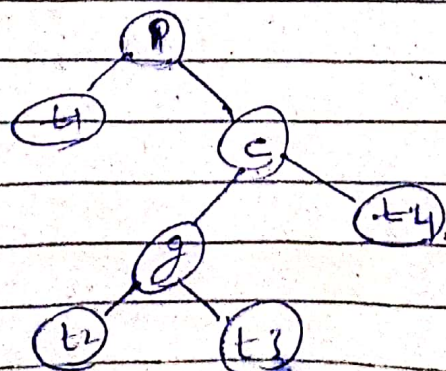
It is Balanced now

3

Right Left Case

→ Child is at right side of parent.  
grandchild is at left side of child.

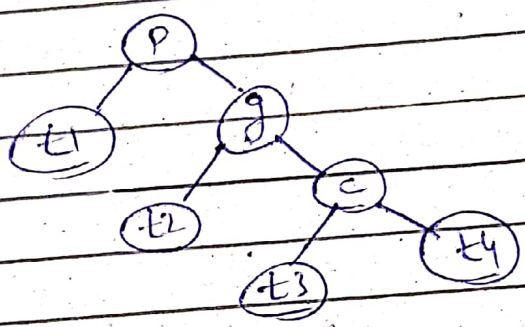
Representation



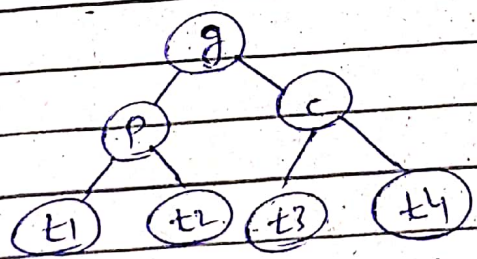


Operations Performed in this case  
 ↳ Right Rotate (c)  
 ↳ Left Rotate (p)

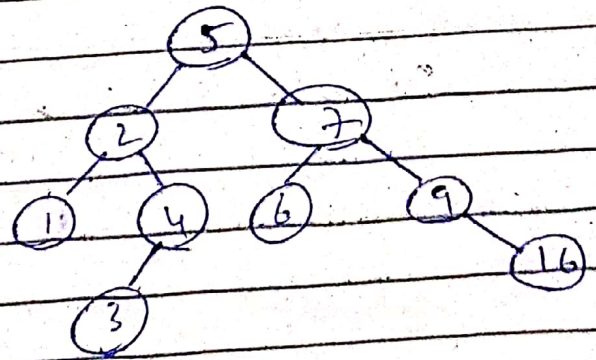
Results  
 ↳ After Right Rotating (c)



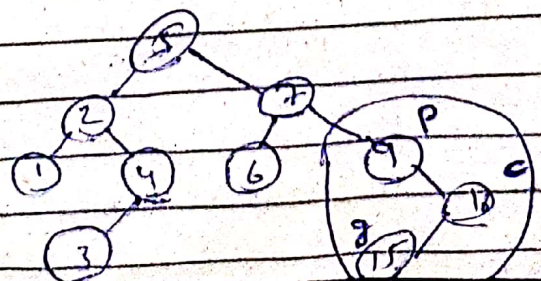
↳ After Left Rotating (p)



↳ Example

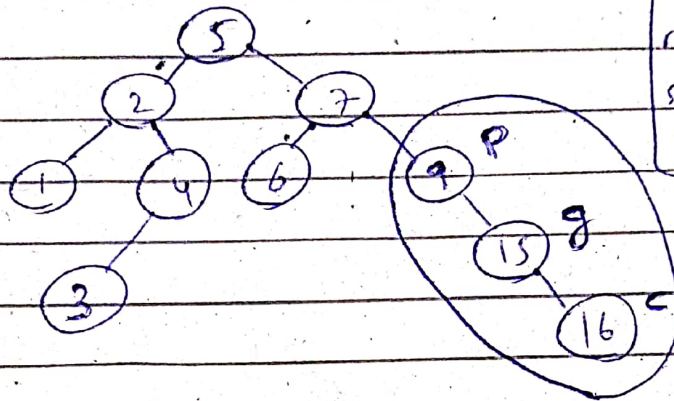


Adding 15 into it



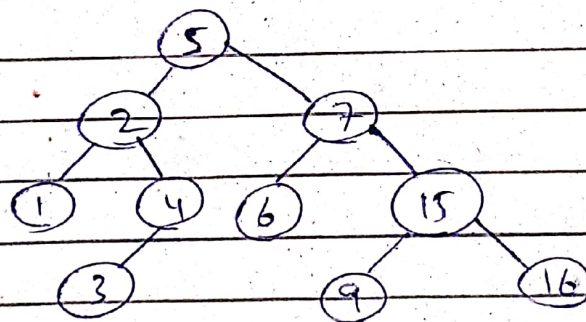
## Applying Operations

### 1. Right Rotating (c)



Note  
Again following  
right-right case  
so left rotating  
P

### 2. Left Rotating (p)

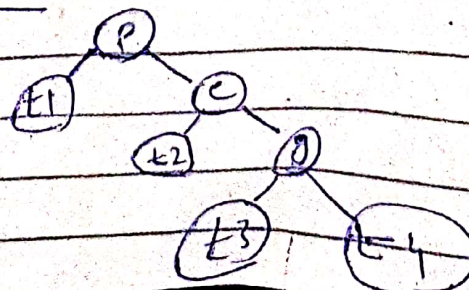


It is now Balanced

### ④ Right Right Case

→ Child is at right side of parent  
Grand Child is at right side of child

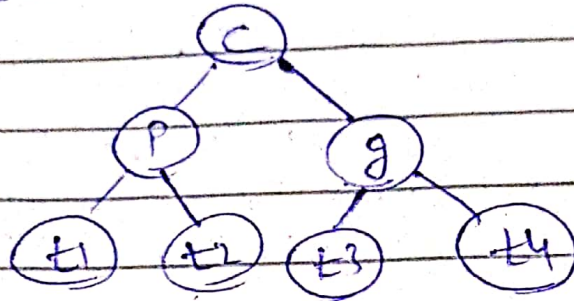
→ Representation :-



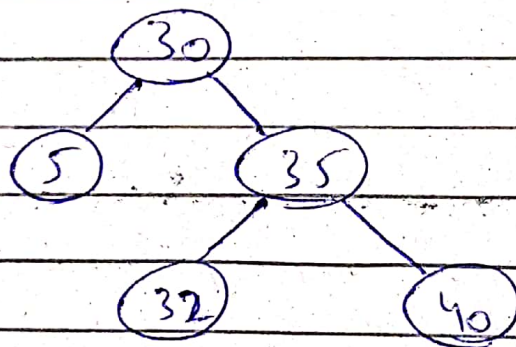


Operations Performed in the case  
Left Rotate (p)

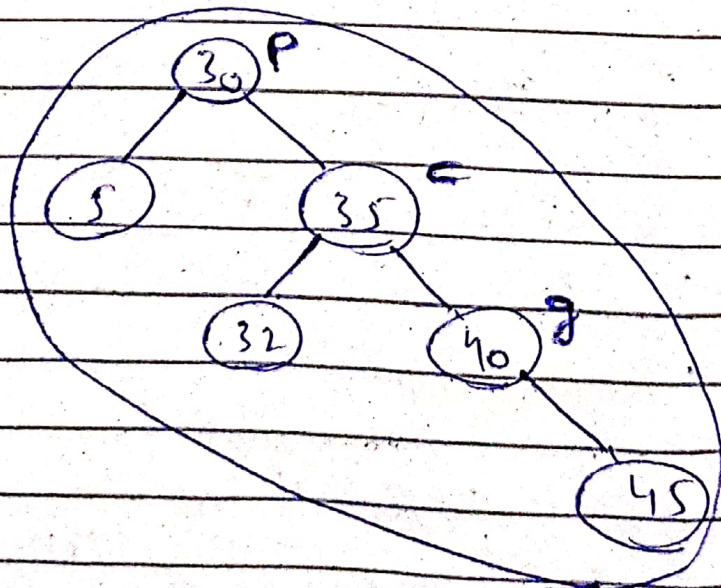
Results



Example

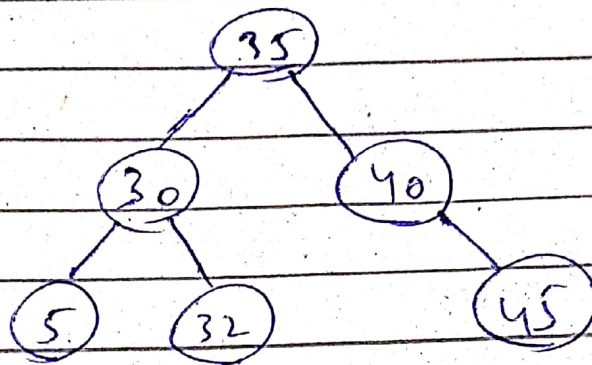


Adding 45 into this



## Applying Operation

Left Rotating (p)



It is now Balanced

## \* Time Complexity

Adding

$$\log(N) + O(1)$$

↓  
Rotation

$$O(\log(N))$$

Because the tree is  
balanced every time.