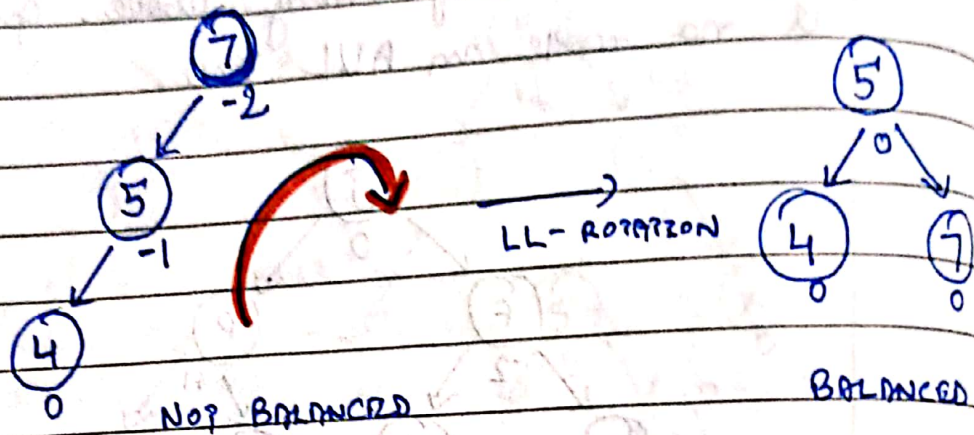


Insertion & Rotation in AVL Trees: 80

① LL Rotation:

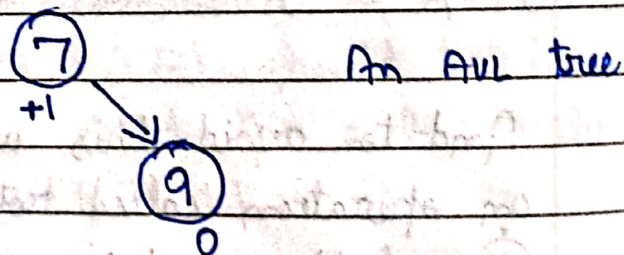
→ The name LL, just because we inserted the new element to the left subtree

of the root. In this rotation technique, you just simply rotate your tree one time in the clockwise direction as shown:

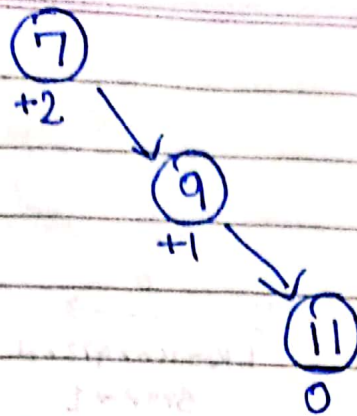


② RR Rotation:

Now, suppose we have a small AVL tree having just these two nodes.

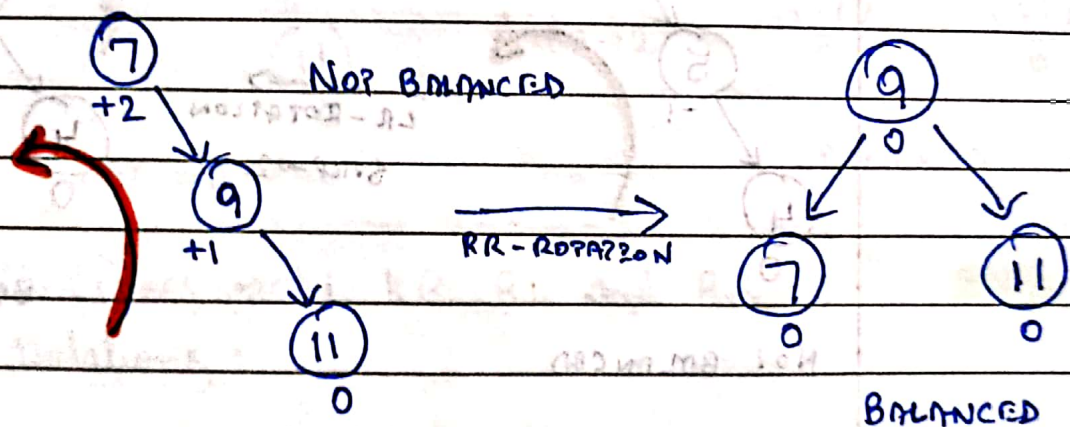


And you see, the balance factor of both the nodes are good, but as soon as we insert a new node having data 11, our updated tree becomes unbalanced to the right. The absolute balance factor of node 7 becomes greater than 1.



Not an AVL anymore

→ Now, we apply RR rotation. The name RR, just because we inserted the new element to the right subtree of the root. In this rotation technique, you just simply rotate your tree one time in an anti-clockwise direction as shown:



③ LR Rotation:

→ The name LR, just because we inserted the new element to the right to the left subtree of the root. In this rotation technique, there is a subtle complexity, which says, first rotate the left subtree in the anticlockwise direction, and then the whole tree in the clockwise direction.

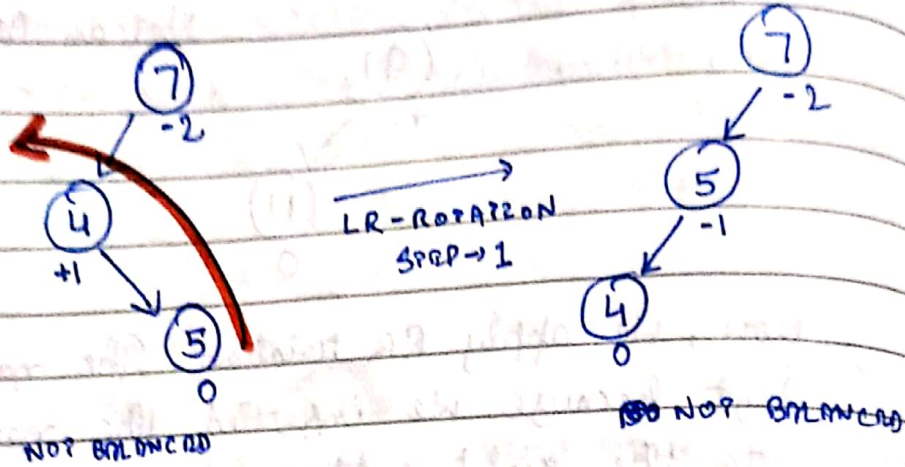
$$BF = R_N - L_N$$

Node with 0 child = 0

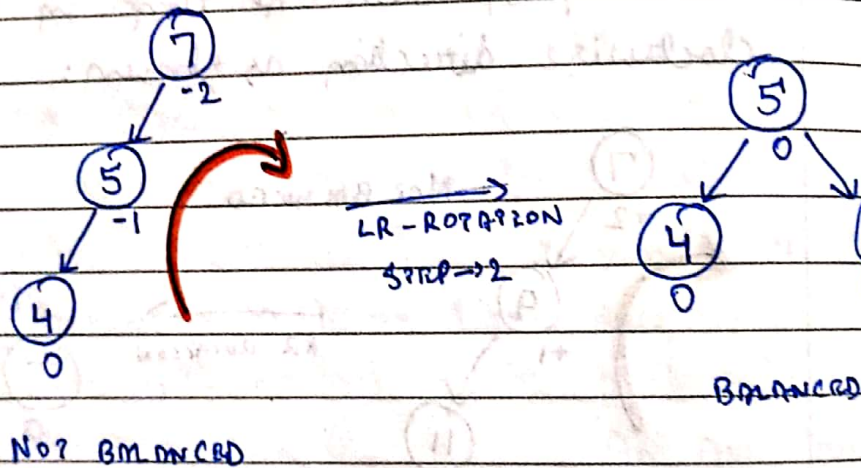
Node No 1

NO NODE = -1

STEP 1:



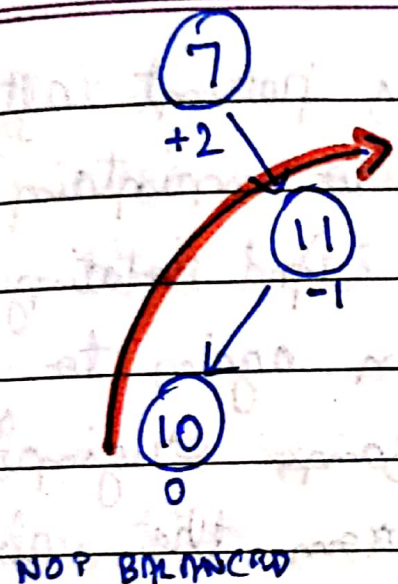
STEP 2:



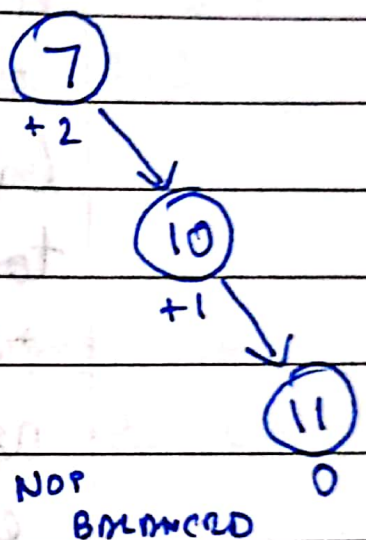
④ RL Rotation:

→ The name RL, just because we inserted the new element to the left to the right subtree of the root, first rotate the right subtree in the clockwise direction, and then the whole tree in the anticlockwise direction.

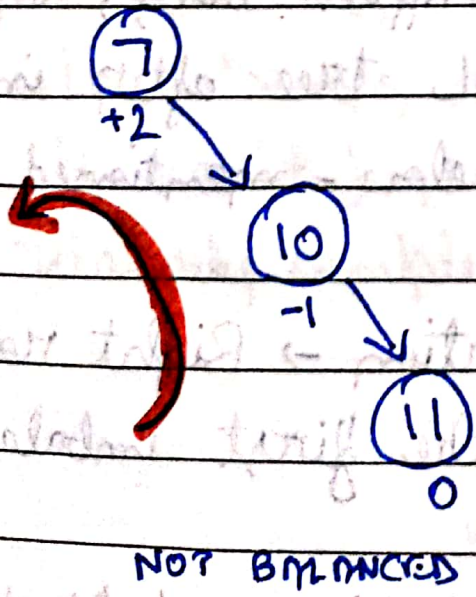
STEP 1:



RL - ROTATION
 STEP → 1



STEP 2:



RL - ROTATION
 STEP → 2

