Lab07: Learn to build/draw AVL tree and understand different types of rotations performed while constructing an AVL tree.

By: Javaid Iqbal

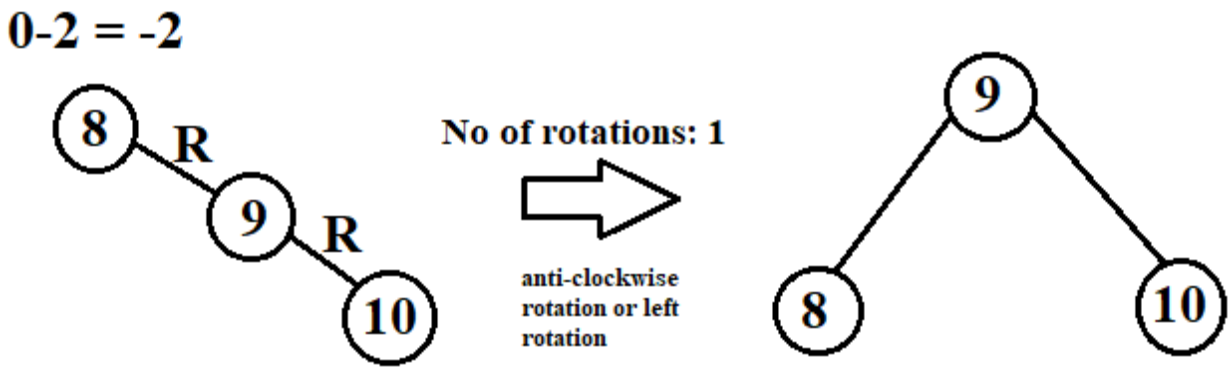
AVL Tree ➔ Balanced Binary Search Tree (BST)

Balance Factor BF = Left Subtree (LST) – Right Subtree (RST)

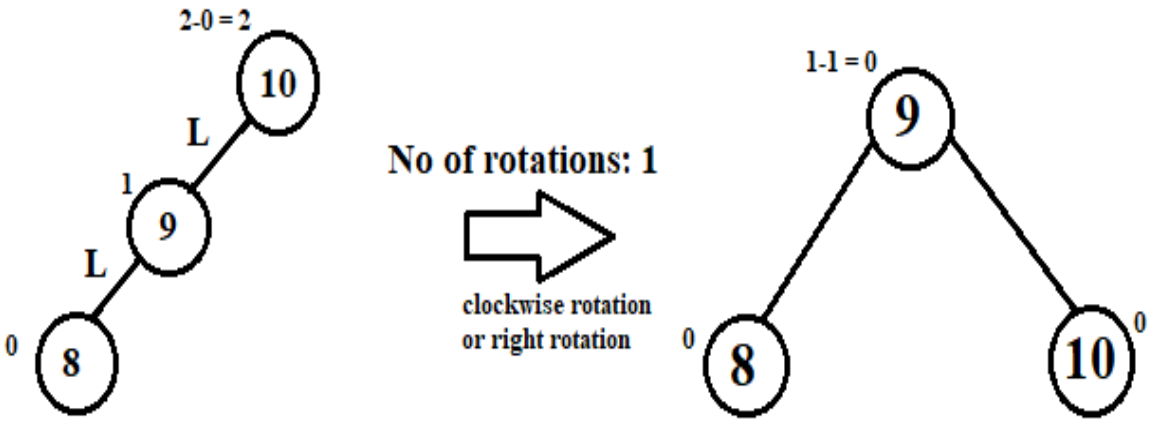
Balance factors: [0, 1, -1]

Cases of Un-balanced Subtree:

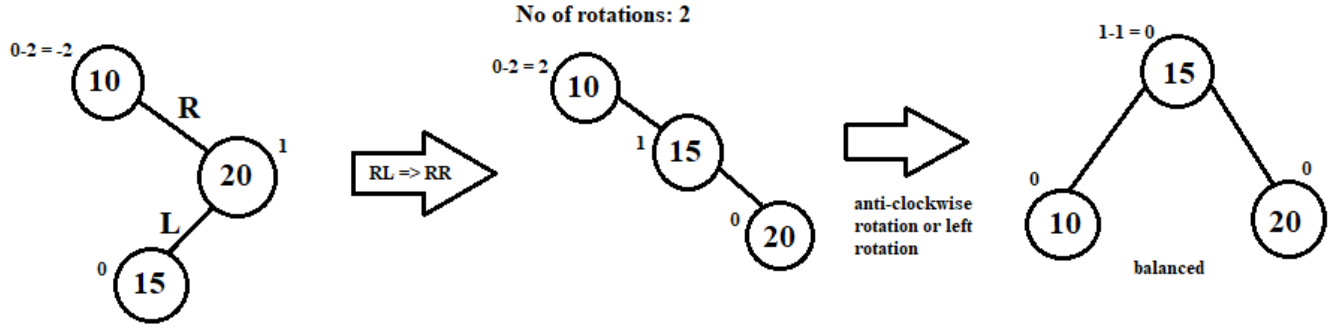
1. RR Unbalanced



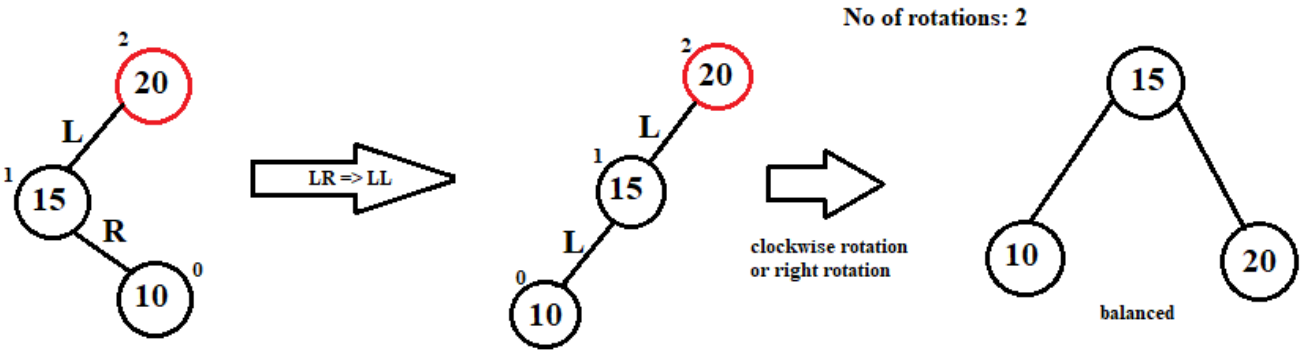
2. LL Unbalanced



3. RL Unbalanced



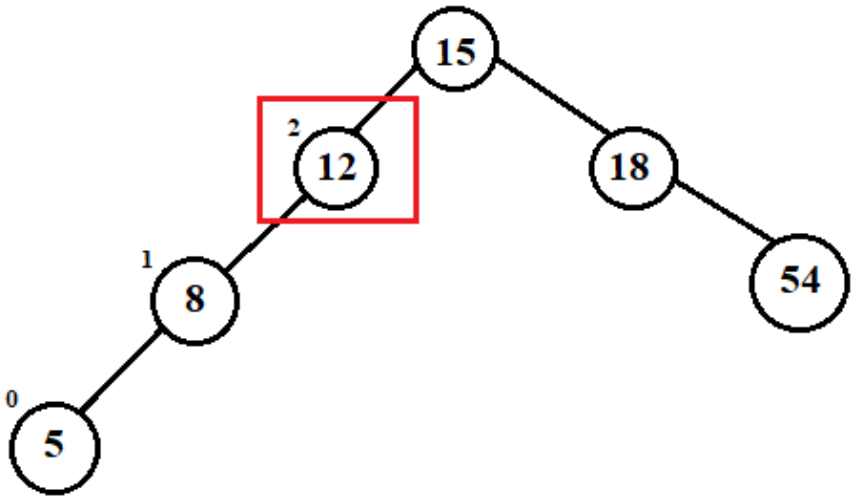
4. LR Unbalanced



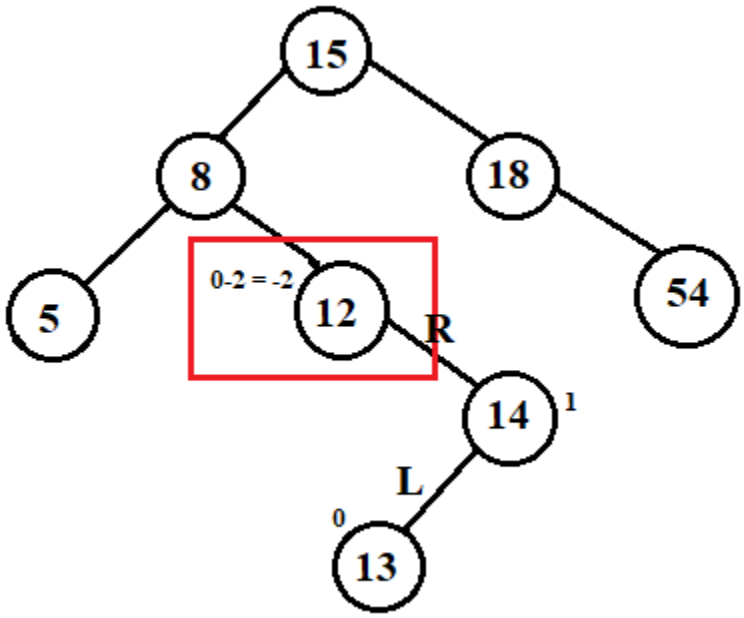
You are required to construct AVL tree from the following data:

15, 18, 12, 8, 54, 5, 14, 13, 9, 61, 20, 17, 21

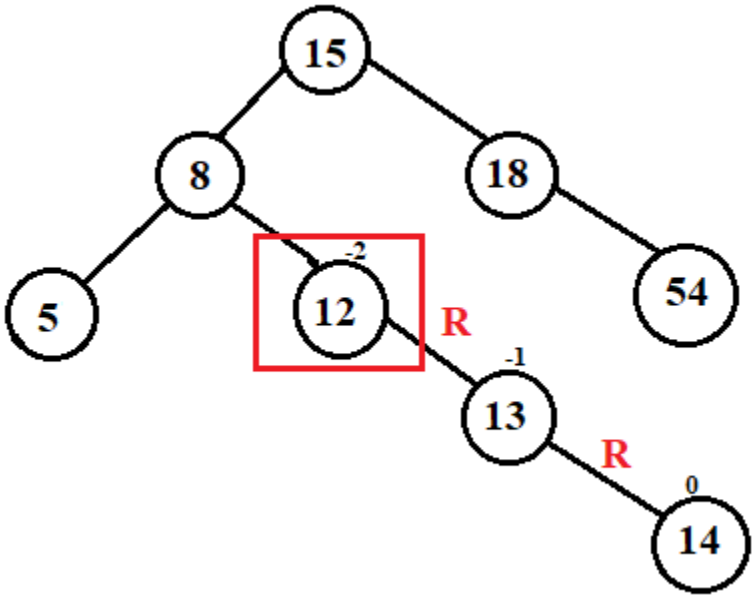
Solution:



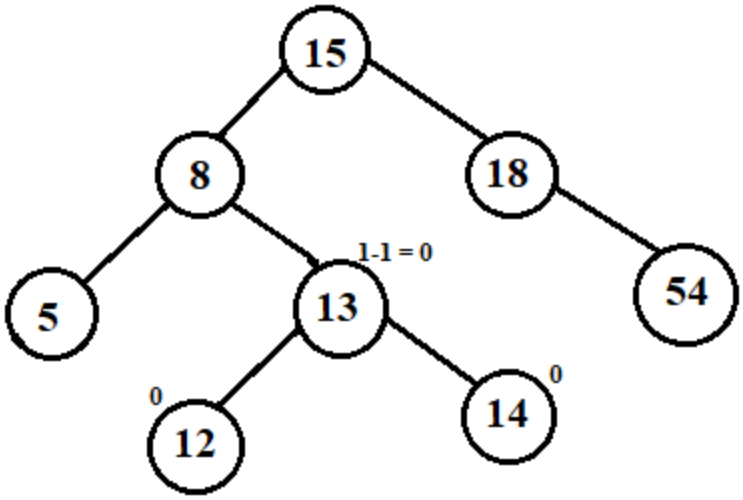
Inserted 15, 18, 12, 8, 54 in AVL tree, tree remains balance but when we inserted 5. At 12 tree becomes LL Unbalanced, so to balance it we apply clockwise rotation.



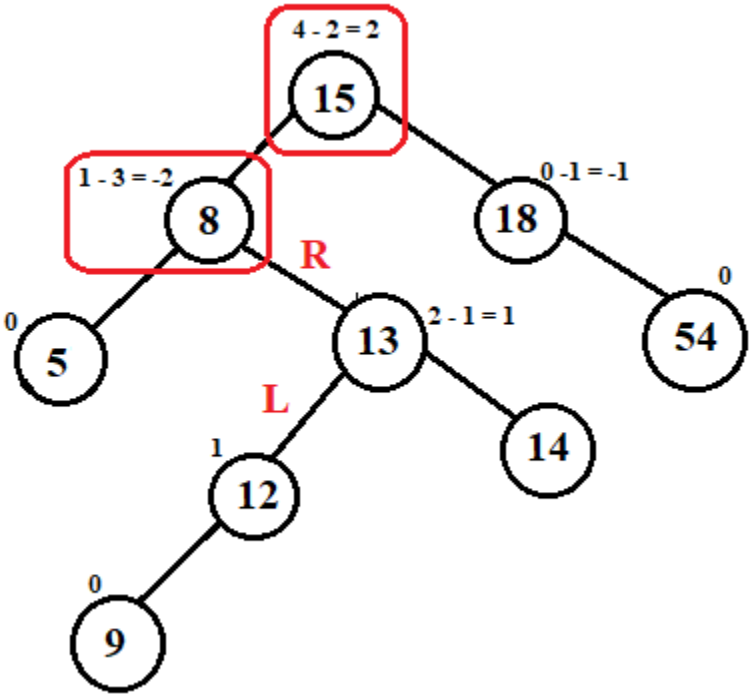
Now again after inserting 14 and 13 tree become RL Unbalanced so we first do RR rotation



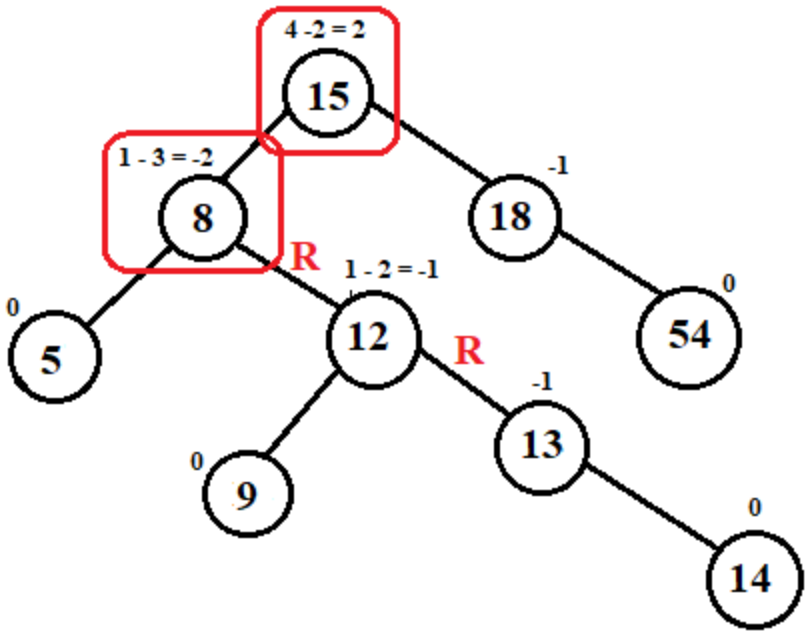
Now still tree is RR Unbalanced, so we apply anti-clockwise rotation



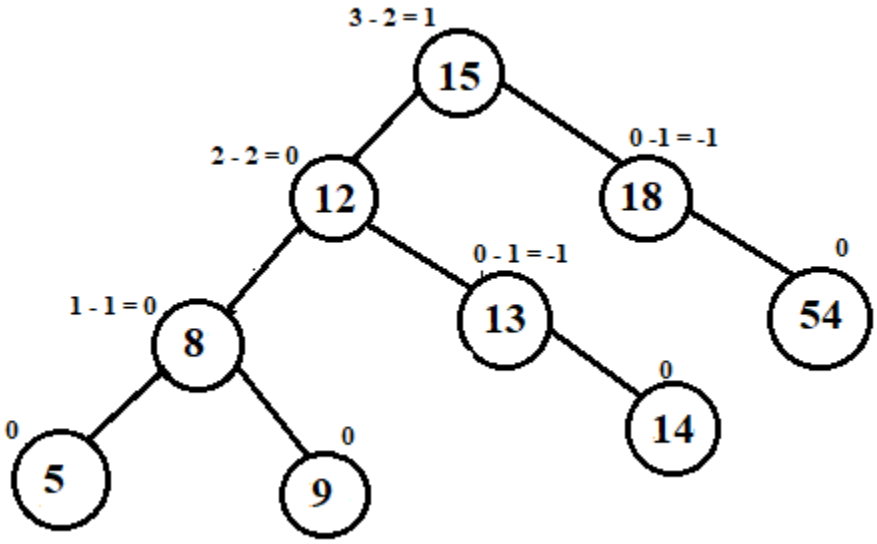
Now our tree is balance, let's insert next nodes.



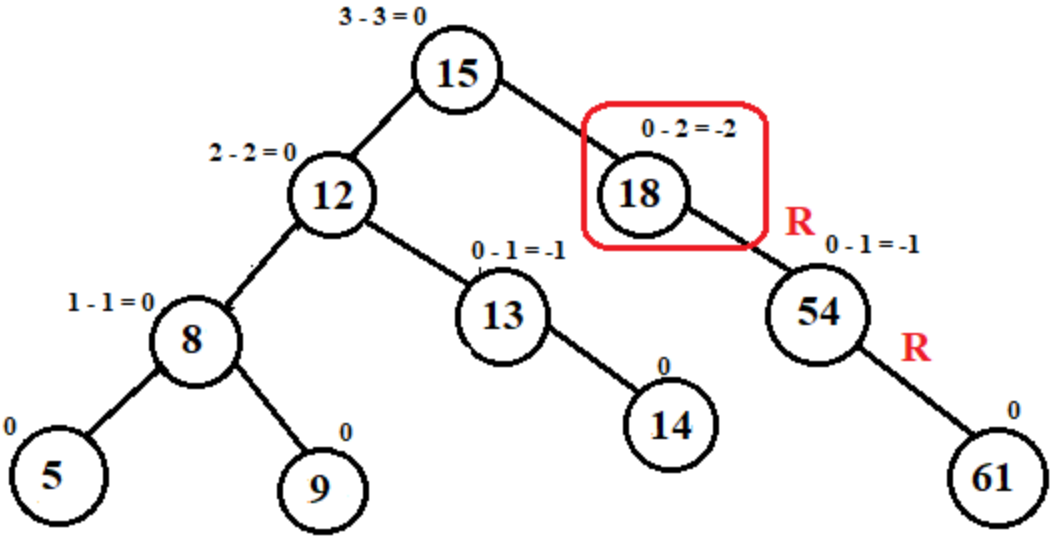
After inserting 9 our tree becomes RL Unbalanced, at 8 so we first apply RR rotation



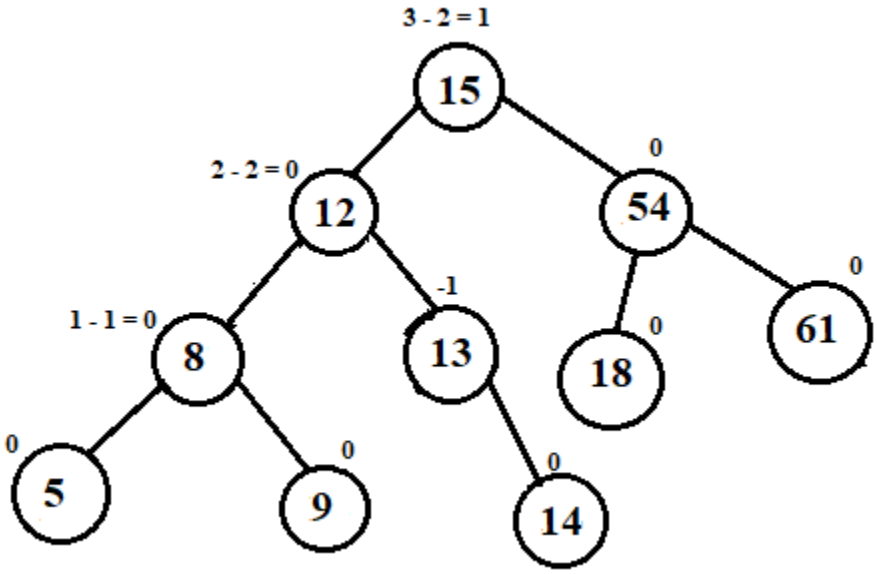
Now still it is unbalanced now we do left or anti-clockwise rotation.



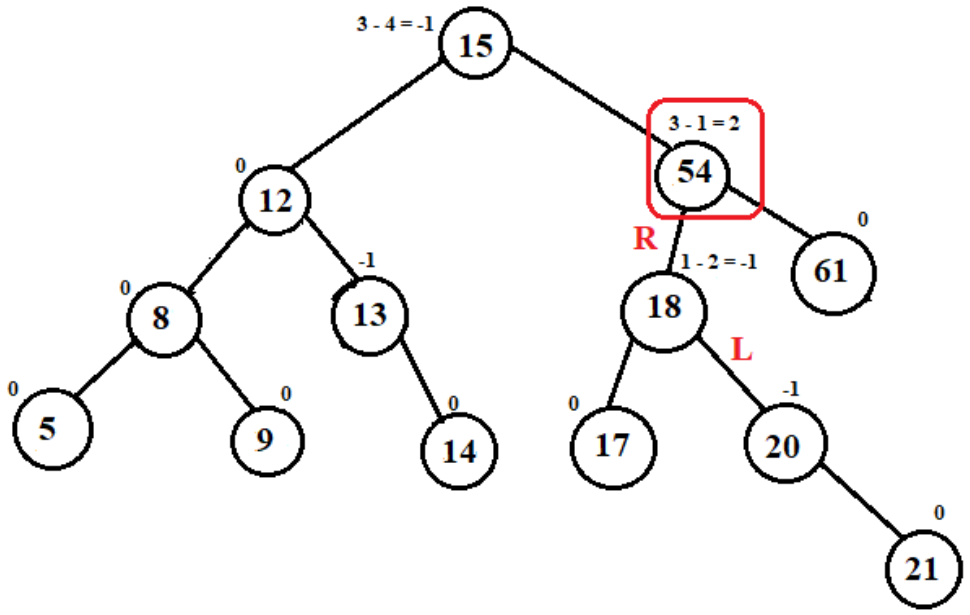
Now our tree is balanced, so let's insert next nodes.



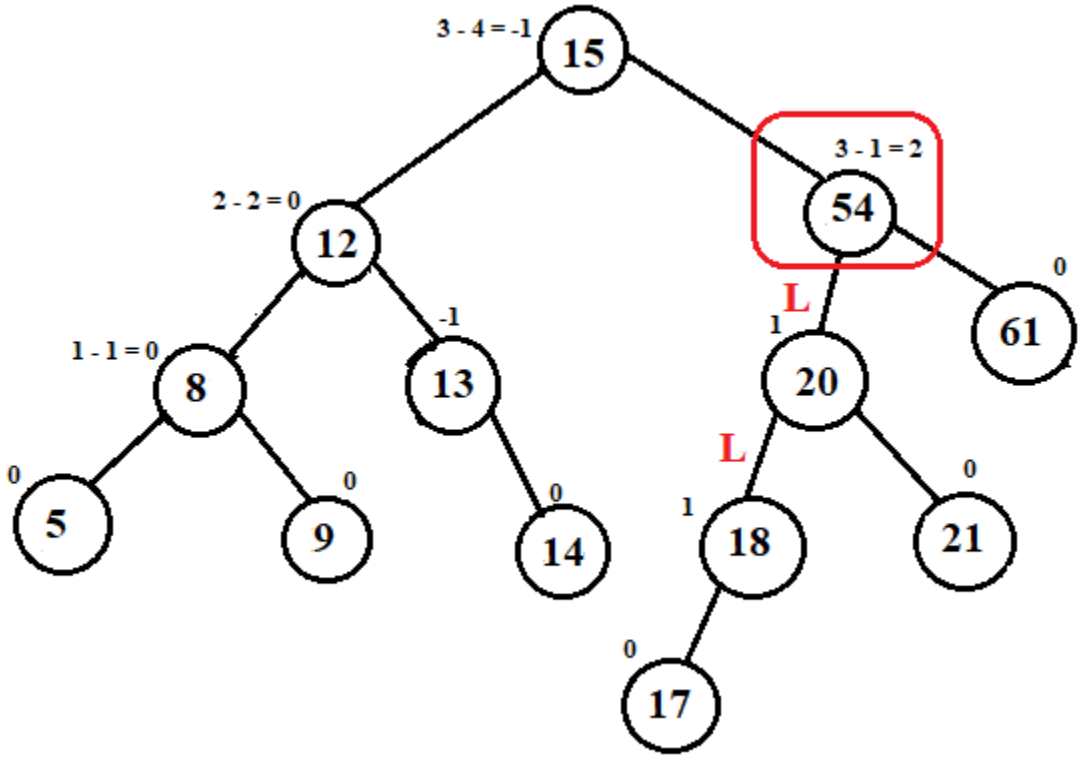
After inserting 61 our tree is RR Unbalanced at 18, so we apply anti-clockwise rotation.



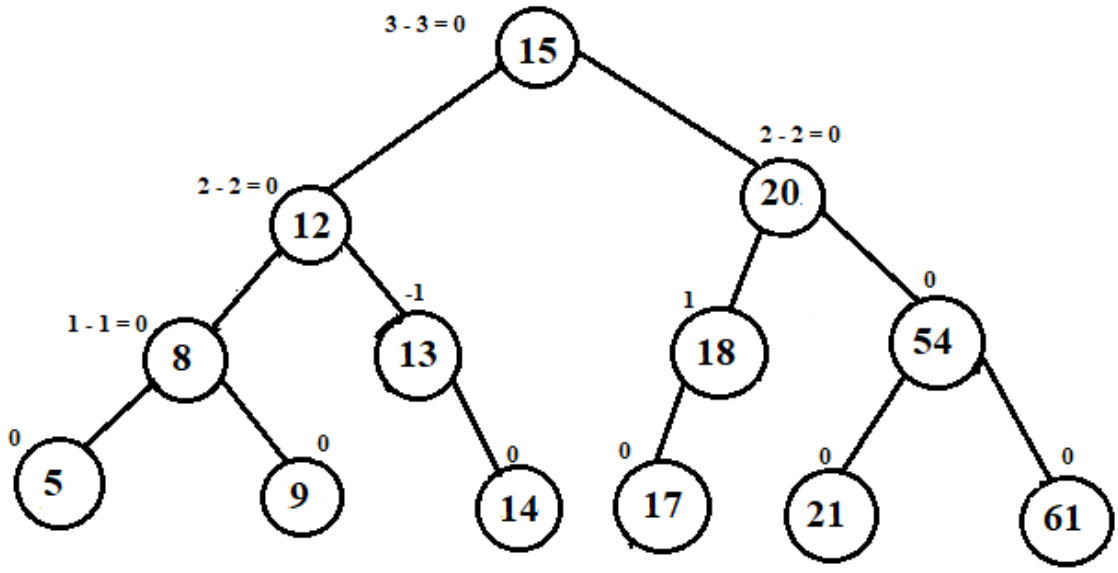
Now our tree is balanced, let's insert next nodes.



After inserting 20, 17 and 2, our tree becomes RL Unbalanced at 54. So we first apply RR rotation.



Now we next do clock-wise rotation.



This is our final AVL Tree.