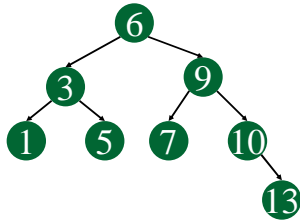


AVL Tree Problem

- Given the following AVL Tree, performs these consecutive operations and draw out the tree in each step:

- Remove(7)
- Insert (11)
- Insert(12)



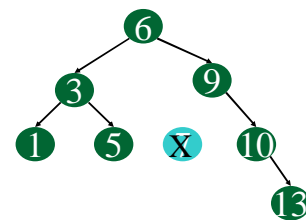
AVL Trees

- AVL Trees are just Binary Search Trees that can rotate their nodes to try to maintain balance.
 - Two kinds of rotations – single and double
 - Can decide which to do based on structure of tree

Insertions/Removals

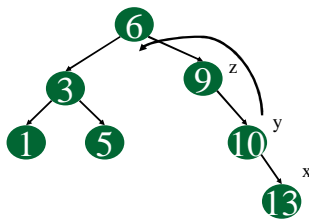
- You have 3 nodes of importance, which we will call x, y, and z (z is the parent of y which is the parent of x)
 - If x is the right child of y, and y is the right child of z, you do a single rotation (same goes for left child of left child)
 - If x is the right child of y, and y is the left child of z, you do a double rotation (same goes for left child of right child)

Remove(7)



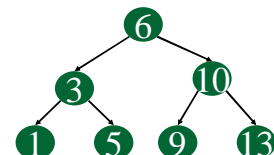
Remove 7 as in BST

Remove(7)



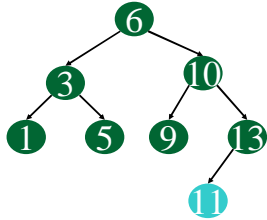
Single rotate

Remove(7)



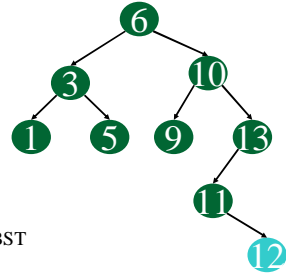
Final tree

Insert(11)



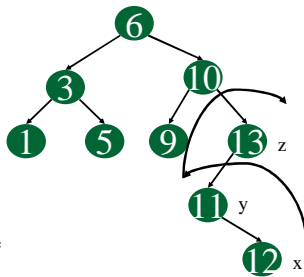
Insert as in BST

Insert(12)



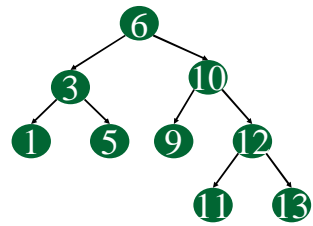
Insert as in BST

Insert(12)



Double rotate

Insert(12)



Final tree