Approach

when inserting a node into an AVL tree, we initially follow the same process as inserting into a Binary search tree (BST).

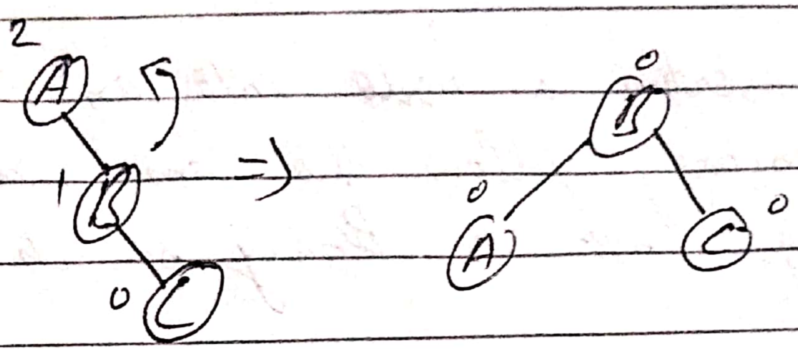
After this insertion if tree becomes unbalanced i.e (height of left sub tree) - (height of right sub tree) = -1, 0, 1 is not followed

To balance itself, an AVL tree may perform the following four kinds of rotations:-

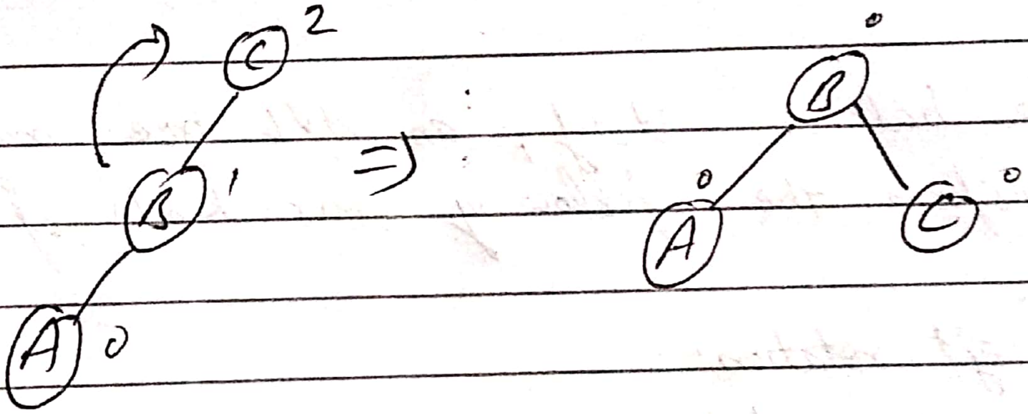
1) Left rotation.
2) Right rotation.
3) Left - Right rotation.
4) Right - Left rotation.

The first two rotations are single rotation and the next two rotations are double rotations. To have an unbalanced tree, we at least need a tree of height 2.

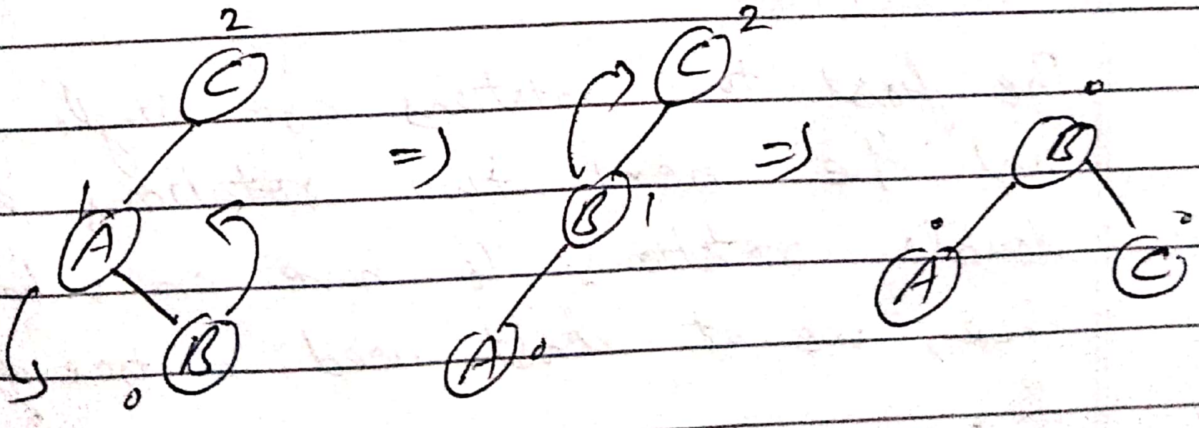
Left Rotation :-



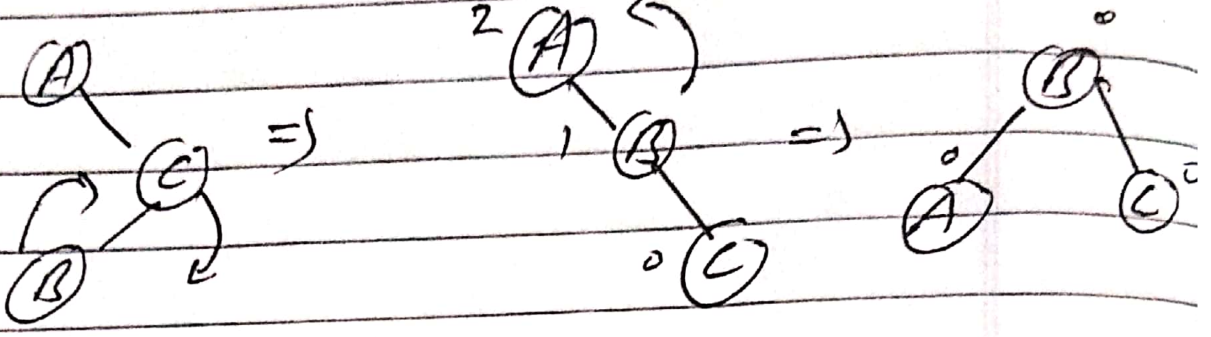
Right Rotation :-



Left - Right Rotation :-



Right - Left Rotation :-



1) If Balance factor (left Subtree height - Righ Subtree height) is greater than 1, then the case is either ~~used~~ ~~have~~ Left left case or Left Right case. To check whether it is left left case or not, compare newly inserted key with the key in left subtree root.

2) If Balance factor is -1, then the case is either Right right or Right left. To check whether it is Right right case compare newly inserted key with the key in right subtree root.