Sairam INSTITUTIONS

Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

YEAR	SEM
II	III

CS8391

DATA STRUCTURES
(Common to CSE & IT)

UNIT - III

NON LINEAR DATA STRUCTURES - TREES

3.5 AVL TREE

- 3.5.1. Insertion - Single Rotation
- 3.5.2. Insertion - Double Roatation
- 3.5.3. Deletion

4 QUALITY EDUCATION

NBA NATIONAL BOARD OF ACCREDITATION

NAAC (A+)

nirf

AVL Tree

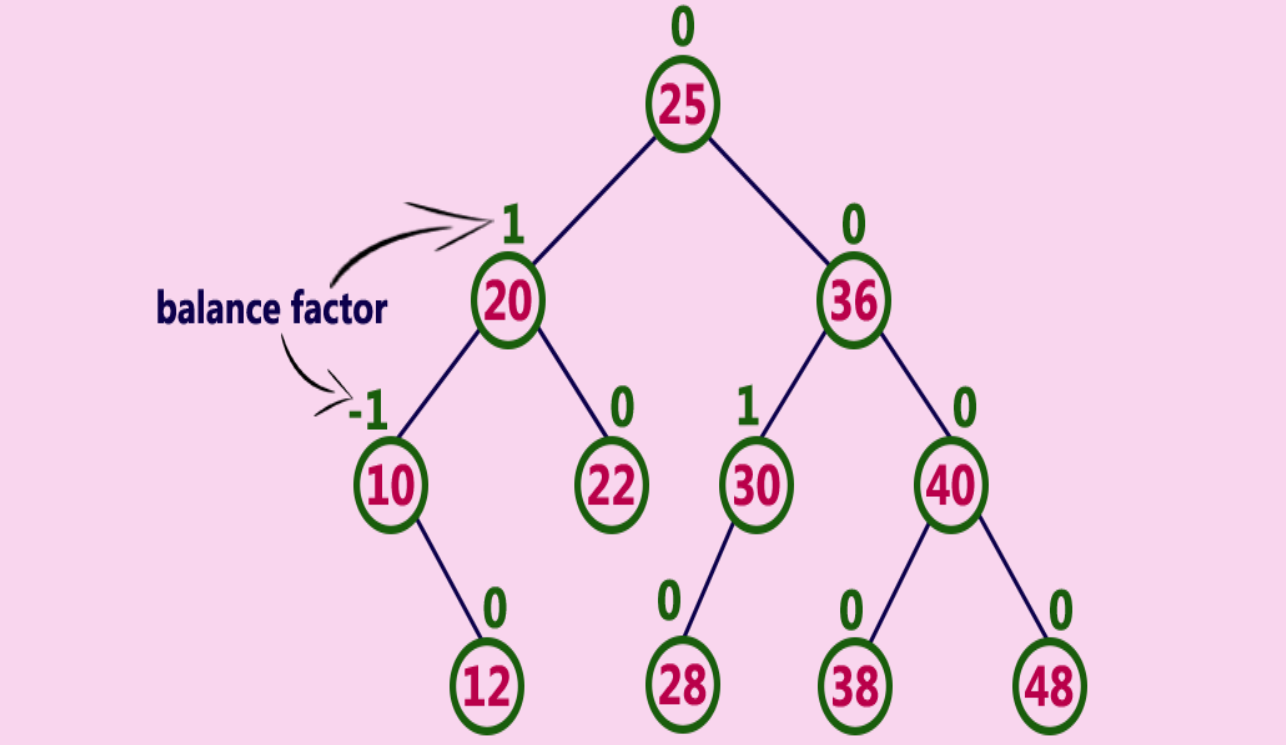
AVL tree is a height-balanced binary search tree. That means, an AVL tree is also a binary search tree but it is a balanced tree. A binary tree is said to be balanced if the difference between the heights of left and right subtrees of every node in the tree is either -1, 0 or +1. In other words, a binary tree is said to be balanced if the height of left and right children of every node differ by either -1, 0 or +1. In an AVL tree, every node maintains an extra information known as **balance factor**.

AVL tree Definition: An AVL tree is a balanced binary search tree. In an AVL tree, the balance factor of every node is either -1, 0 or +1.

Balance factor of a node is the difference between the heights of the left and right subtrees of that node. The balance factor of a node is calculated either height of left subtree - height of right subtree.

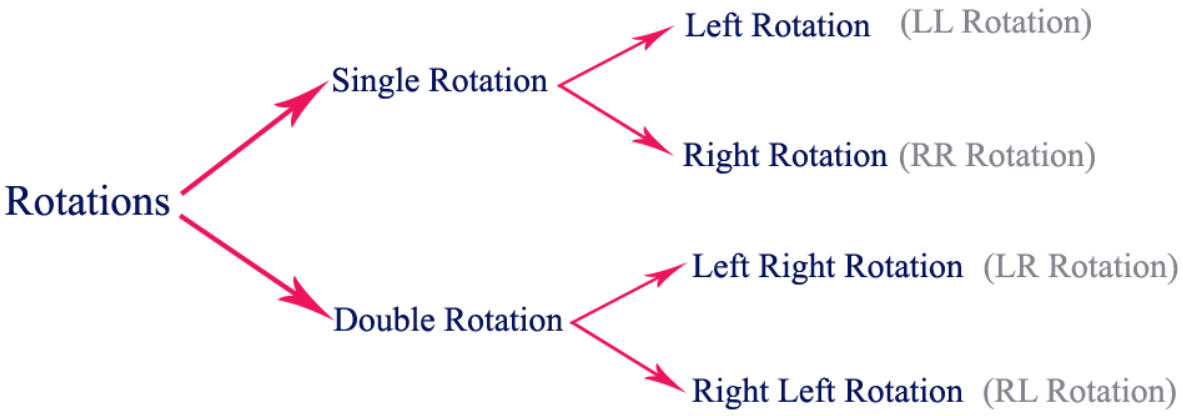
Balance factor = heightOfLeftSubtree - heightOfRightSubtree

Example of AVL Tree



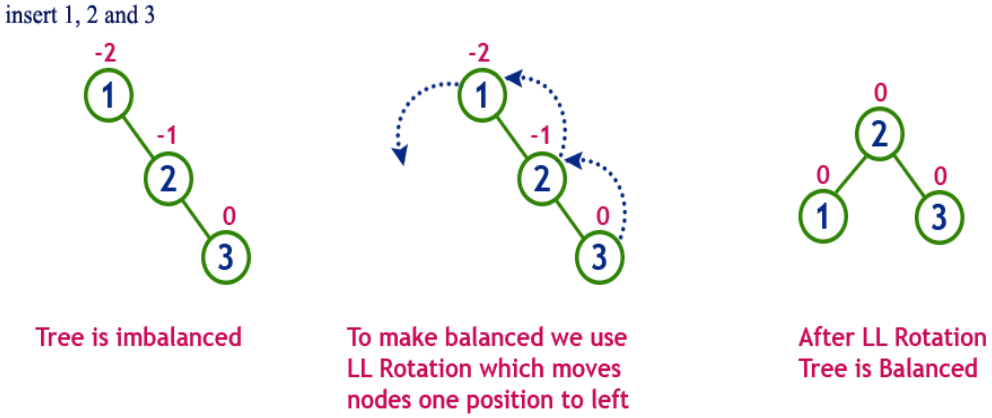
AVL Tree Rotations

Rotation is the process of moving nodes either to left or to right to make the tree balanced. There are four rotations and they are classified into two types.



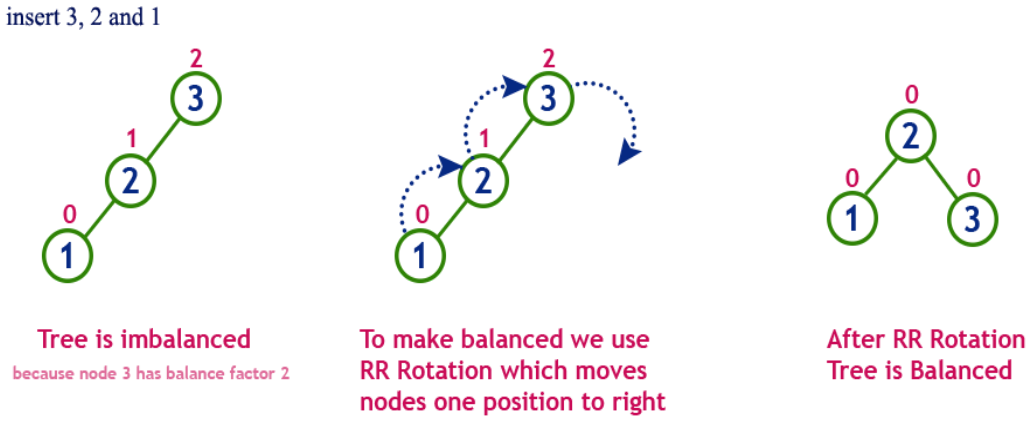
Single Left Rotation (LL Rotation)

In LL Rotation, every node moves one position to left from the current position. To understand LL Rotation, let us consider the following insertion operation in AVL Tree.



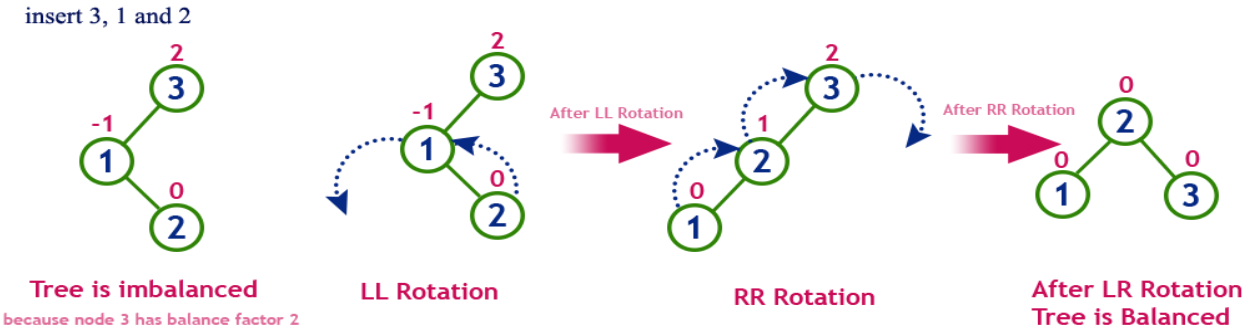
Single Right Rotation (RR Rotation)

In RR Rotation, every node moves one position to right from the current position. To understand RR Rotation, let us consider the following insertion operation in AVL Tree.



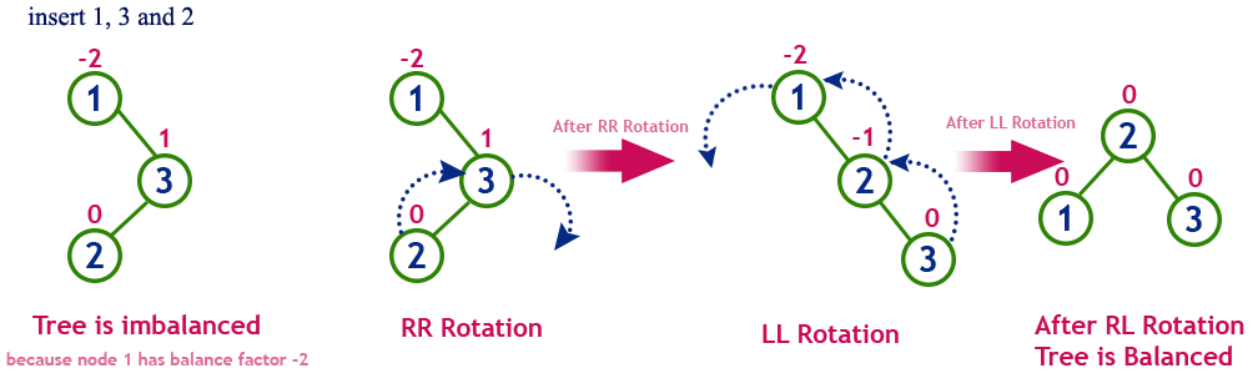
Left Right Rotation (LR Rotation)

The LR Rotation is a sequence of single left rotation followed by a single right rotation. In LR Rotation, at first, every node moves one position to the left and one position to right from the current position. To understand LR Rotation, let us consider the following insertion operation in AVL Tree.



Right Left Rotation (RL Rotation)

The RL Rotation is a sequence of single right rotation followed by single left rotation. In RL Rotation, at first every node moves one position to right and one position to left from the current position. To understand RL Rotation, let us consider the following insertion operation in AVL Tree.



Operations on an AVL Tree

The following operations are performed on AVL tree.

1. Insertion
2. Deletion

Insertion Operation in AVL Tree

In an AVL tree, the insertion operation is performed with O(log n) time complexity. In AVL Tree, a new node is always inserted as a leaf node. The insertion operation is performed as follows.

Step 1 - Insert the new element into the tree using Binary Search Tree insertion logic.

Step 2 - After insertion, check the Balance Factor of every node.

Step 3 - If the Balance Factor of every node is 0 or 1 or -1 then go for the next operation.

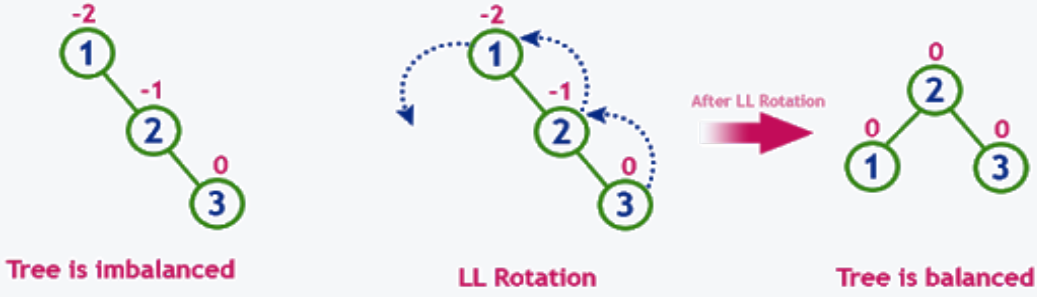
Step 4 - If the Balance Factor of any node is other than 0 or 1 or -1 then that tree is said to be imbalanced. In this case, perform suitable Rotation to make it balanced and go for the next operation.

Example: Construct an AVL Tree by inserting numbers from 1 to 8.

insert 1

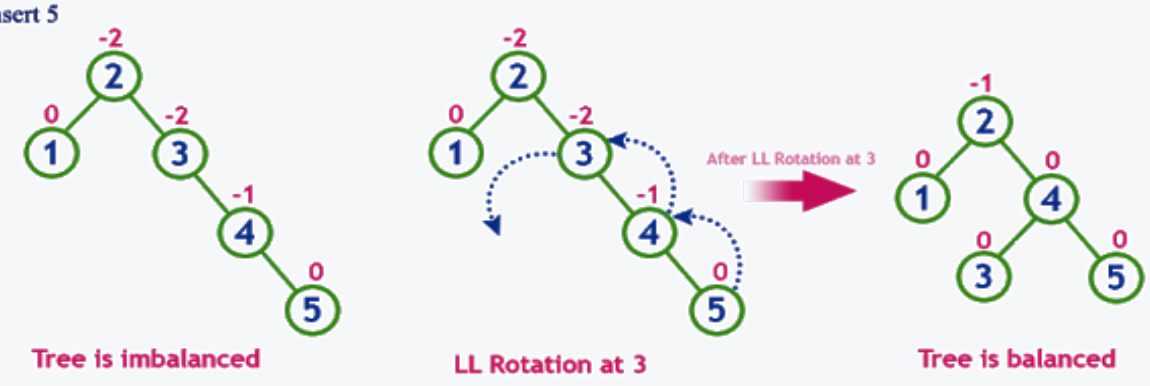
insert 2

insert 3



insert 4

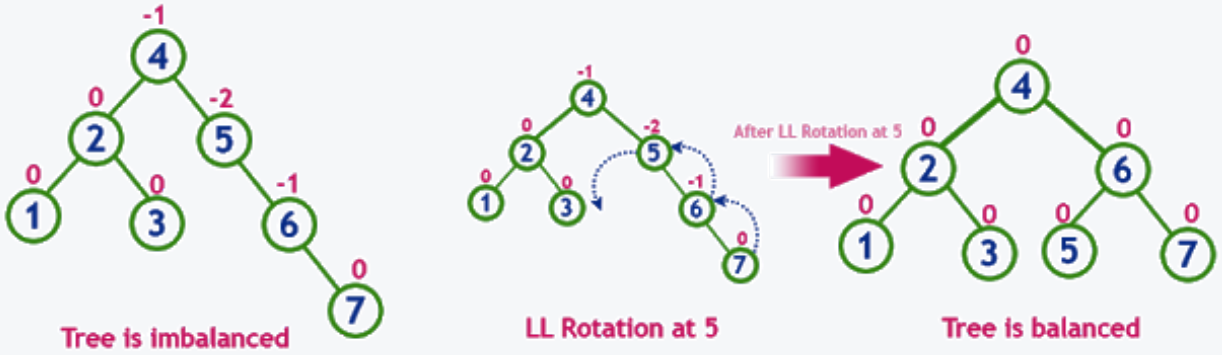
insert 5



insert 6

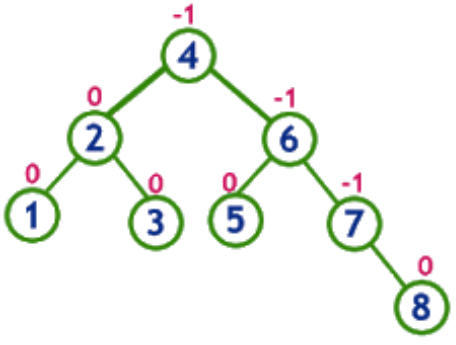
Tree is imbalanced | LL Rotation at 2 | Tree is balanced

insert 7



Tree is imbalanced | LL Rotation at 5 | Tree is balanced

insert 8



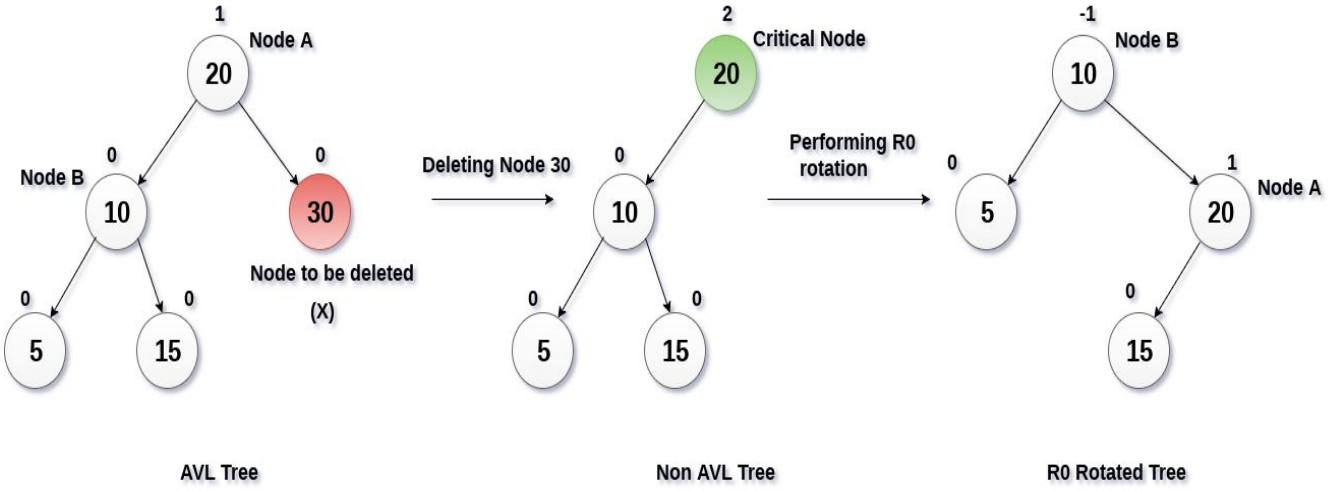
Tree is balanced

Deletion Operation in AVL Tree

The deletion operation in AVL Tree is similar to the deletion operation in BST. But after every deletion operation, we need to check with the Balance Factor condition. If the tree is balanced after deletion go for the next operation otherwise perform suitable rotation to make the tree Balanced.

Example1:

Delete the node 30 from the AVL tree shown in the following image.



Example2:

Delete the node 30 from the AVL tree shown in the following image.