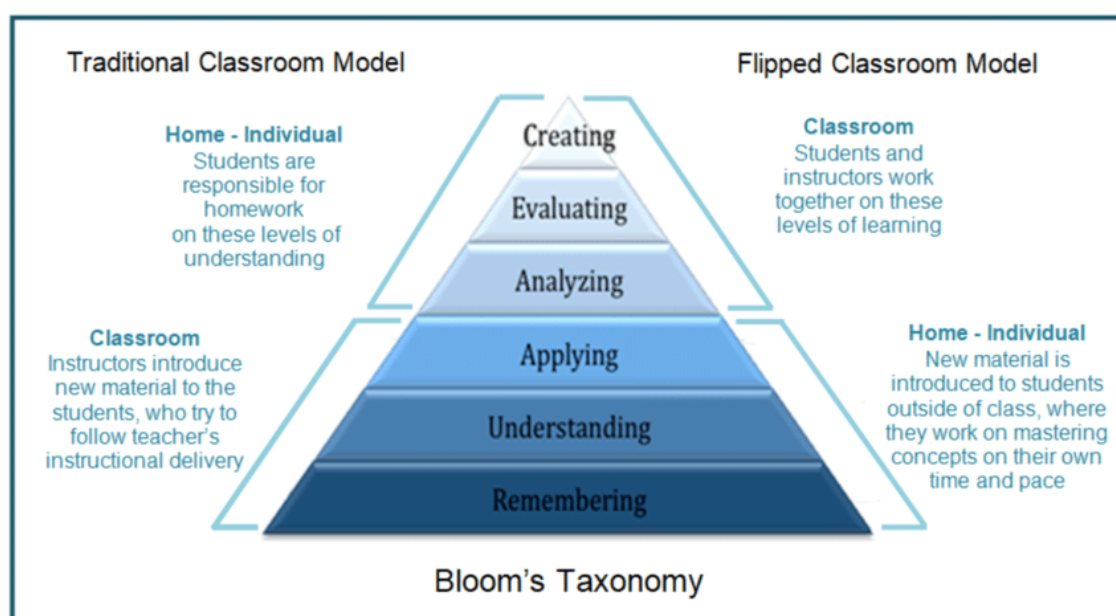


Introduction

Flipping the classroom is a simple concept. While a traditional model involves students learning new material during class time and then practicing it at home, a flipped classroom reverses things so that students learn new material at home—typically by watching a video or other presentation—and then practice during class time. There are two main advantages to flipping the classroom: students seem to be more engaged with their homework when it is a video than when it is a worksheet and, secondly, practice is normally more productive when the teacher is available to help students who need guidance. Of course, this is a radically different approach to instruction, and adopting it requires significant effort on the part of the teacher.



Faculty Details:

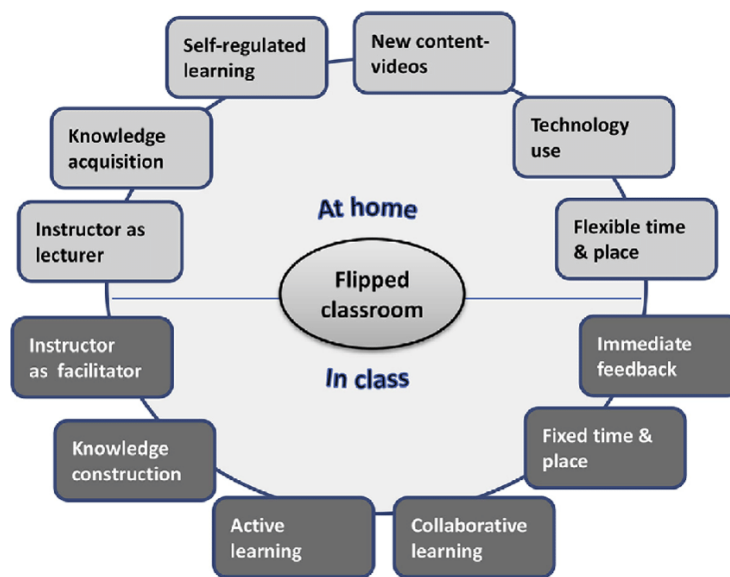
Name Of The Faculty: Ms. Tahera Abid	Designation: Asst Professor	Subject: Data Structure
Course: B.E	Year- Sem : II-III	Topic : AVL Tree

Objective of the Activity:

There are a variety of edtech tools that can help when a classroom is flipped, but Google Classroom may be the best option. One of its most important features is that it makes it easy for faculty to keep

all of their instructional materials in one place. This way, as faculty re-use videos and presentations from class to class or from year to year, it's easier to stay organized.

Another benefit of using Google Classroom to flip lessons is that it integrates materials from Docs, Sheets, Drive, and Gmail, so that it is easy for faculty to make assignments, for students to access the instructional materials that they need, and for faculty to assess student learning. Discussion threads mean that students and faculty can communicate outside of college hours as needed.



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. The AVL tree was introduced in the year 1962 by G.M. Adelson-Velsky and E.M. Landis.

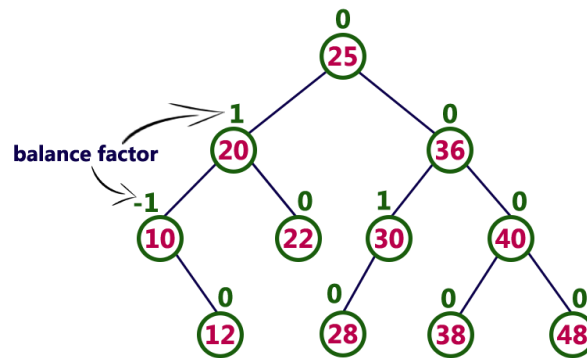
An AVL tree is defined as follows...

An AVL tree is a balanced binary search tree. In an AVL tree, 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 (OR) height of right subtree - height of left subtree. In the following explanation, we calculate as follows...

$$\text{Balance factor} = \text{heightOfLeftSubtree} - \text{heightOfRightSubtree}$$

Example of AVL Tree



The above tree is a binary search tree and every node is satisfying balance factor condition. So this tree is said to be an AVL tree. Every AVL Tree is a binary search tree but every Binary Search Tree need not be AVL tree.

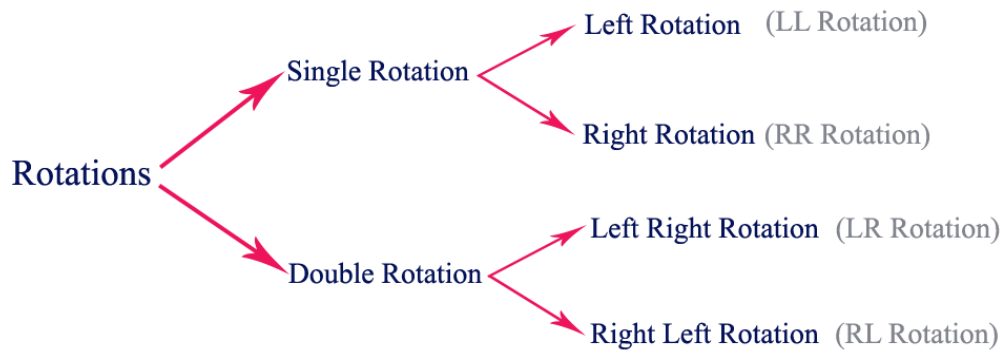
AVL Tree Rotations

In AVL tree, after performing operations like insertion and deletion we need to check the balance factor of every node in the tree. If every node satisfies the balance factor condition then we conclude the operation otherwise we must make it balanced. Whenever the tree becomes imbalanced due to any operation we use rotation operations to make the tree balanced.

Rotation operations are used to make the tree balanced.

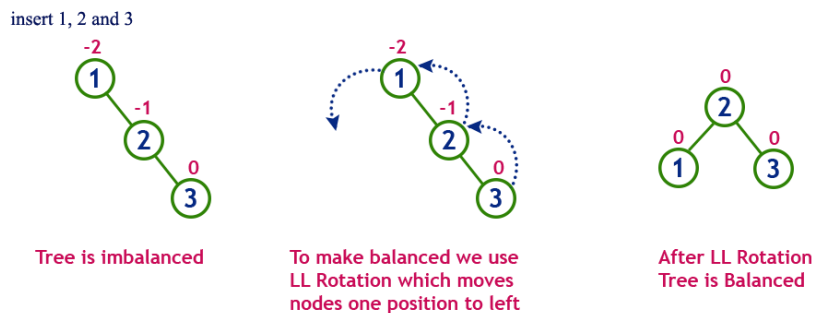
Rotation is the process of moving nodes either to left or to right to make the tree balanced.

There are four rotations and they are classified into two types.



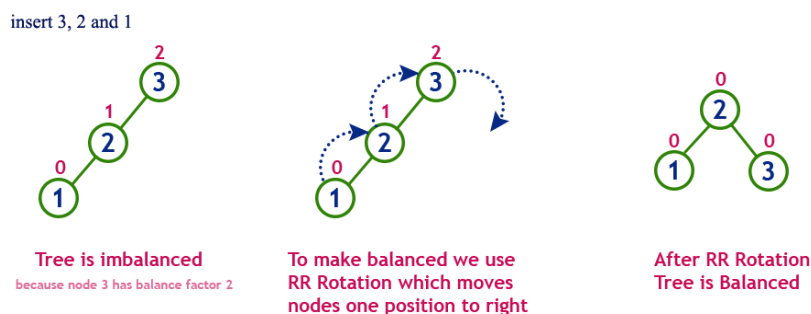
Single Left Rotation (LL Rotation)

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...



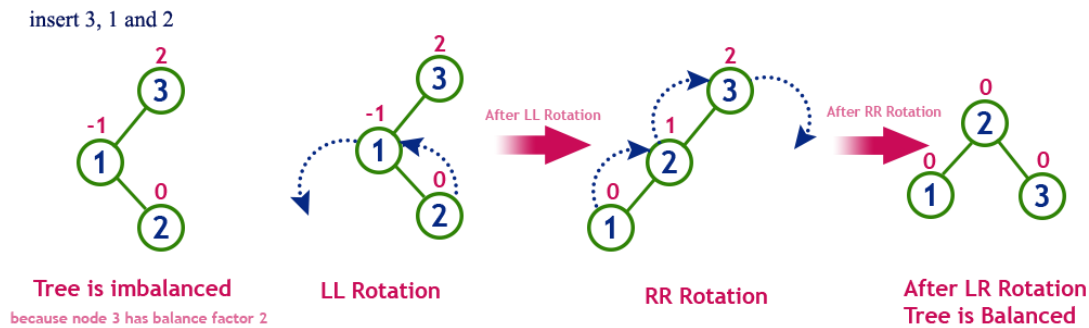
Single Right Rotation (RR Rotation)

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...



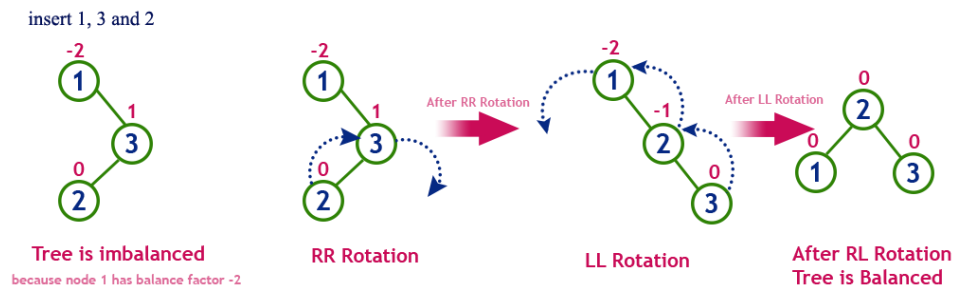
Left Right Rotation (LR Rotation)

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 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...

Search

Insertion

Deletion

Search Operation in AVL Tree

In an AVL tree, the search operation is performed with $O(\log n)$ time complexity. The search operation in the AVL tree is similar to the search operation in a Binary search tree. We use the following steps to search an element in AVL tree...

Step 1 - Read the search element from the user.

Step 2 - Compare the search element with the value of root node in the tree.

Step 3 - If both are matched, then display "Given node is found!!!" and terminate the function

Step 4 - If both are not matched, then check whether search element is smaller or larger than that node value.

Step 5 - If search element is smaller, then continue the search process in left subtree.

Step 6 - If search element is larger, then continue the search process in right subtree.

Step 7 - Repeat the same until we find the exact element or until the search element is compared with the leaf node.

Step 8 - If we reach to the node having the value equal to the search value, then display "Element is found" and terminate the function.

Step 9 - If we reach to the leaf node and if it is also not matched with the search element, then display "Element is not found" and terminate the function.

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 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 next operation.

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

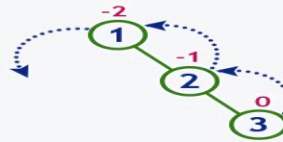
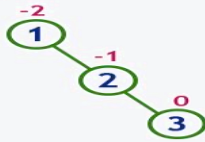
insert 1



insert 2



insert 3



After LL Rotation



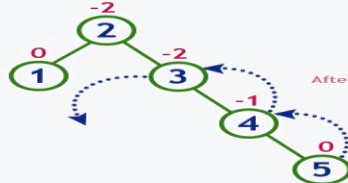
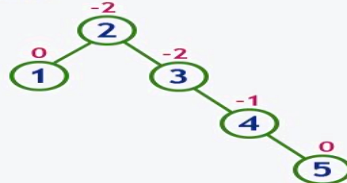
LL Rotation

Tree is balanced

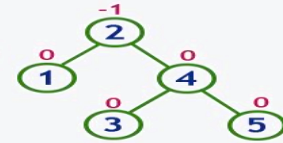
insert 4



insert 5



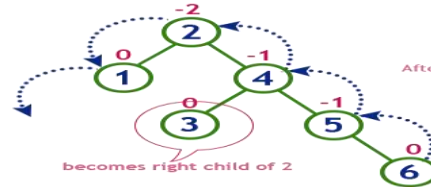
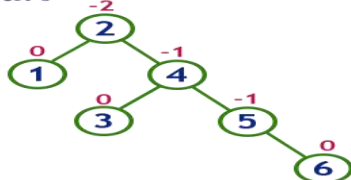
After LL Rotation at 3



LL Rotation at 3

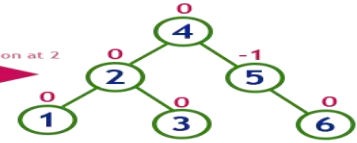
Tree is balanced

insert 6



becomes right child of 2

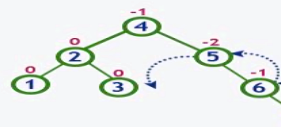
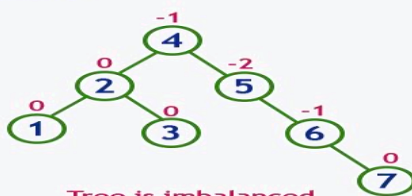
After LL Rotation at 2



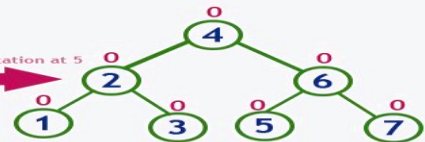
LL Rotation at 2

Tree is balanced

insert 7



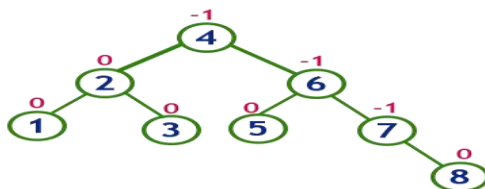
After LL Rotation at 5



LL Rotation at 5

Tree is balanced

insert 8



Deletion Operation in AVL Tree

The deletion operation in AVL Tree is similar to 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 next operation otherwise perform suitable rotation to make the tree Balanced.

Outcomes:

1. The flipped classroom method will increase the knowledge and understanding of the course area.
2. This method makes the topic more engaging than the regular method. It's easier to urge the points from videos than from lecture notes.
3. The chance to observe videos reception, makes it easier for weak students to understand at least the foremost fundamental and important parts of the course.

Activity Flipped Class Room:

Learning Material:

1. <https://www.youtube.com/watch?v=u3OVSkudqI>
2. <https://www.educba.com/avl-tree-in-data-structure/>
3. <https://www.programiz.com/dsa/avl-tree>