**National University of Computer and Emerging Sciences**



Laboratory Manual

for

Data Structures Lab

|  |  |
| --- | --- |
| Course Instructor | Samman Ashraf |
| Lab Instructor(s) | Muhammad Ahmed  Hamza Ayub |
| Section | 4N |
| Semester | Spring 2023 |

**Department of Computer Science**

FAST-NU, Lahore, Pakistan

# AVL Trees

# An AVL tree defined as a self-balancing Binary Search Tree (BST) where the difference between heights of left and right subtrees for any node cannot be more than one.

# The difference between the heights of the left subtree and the right subtree for any node is known as the balance factor of the node. The AVL tree is named after its inventors, Georgy Adelson-Velsky and Evgenii Landis, who published it in their 1962 paper “An algorithm for the organization of information”.

# Example of AVL Trees:

# C:\Users\ahmed\Desktop\avl.png

# The above tree is AVL because the differences between the heights of left and right subtrees for every node are less than or equal to 1.

# 

# To see all the operations of an AVL Tree Visit <https://www.geeksforgeeks.org/introduction-to-avl-tree/> .

# Exercise

**Visualization:** <https://www.cs.usfca.edu/~galles/visualization/AVLtree.html>

# Problem 1:

# Implement a self-balancing tree AVL the structure of the class is given below. You can add helper functions as private data members of the class if required.

# class AVL

# {

# Node\* root;

# int getBalance(Node \*node);

# Node \*leftRotate(Node \*node);

# Node \*rightRotate(Node \*node);

# public:

# void insert(int key);

# void delete(int key);

# void preOrderPrint() const;

# };

# Problem 2:

# Test your implementation by inserting the following values into the tree in the order given: 10, 20, 30, 40, 50, 25, 60.

# Print out the resulting AVL tree after each insertion. Be sure to include the balance factor for each node.

# Implement a function that deletes a node from the AVL tree.

# Test your deletion function by deleting the node with value 25 from the tree.

# Print out the resulting AVL tree after the deletion. Be sure to include the balance factor for each node.

# Calculate the height of the tree after the deletion.