St. Francis Institute of Technology, Mumbai-400 103

Department of Information Technology

A.Y. 2023-24
Class: SE-ITA/B, Semester: III
Subject: DATA STRUCTURE LAB

Experiment – 4: AVL tree operations

1. **Aim:** Write a C program to implement an AVL tree.

2. **Objectives:** After study of this experiment, the student will be able to • To use basic principles of programming as applied to complex data structures • To implement the tree through programming

3. **Outcomes:** After study of this experiment, the student will be able to
   - Formulate and solve problems of AVL trees and its operations
   - Understand the concepts and apply the methods in basic trees

4. **Prerequisite:** Types of trees, Binary tree.

5. **Requirements:** PC and Turbo C compiler version 3.0

6. **Pre-Experiment Exercise:**

Brief Theory:

<u>Explain the following terms</u>

A. AVL Tree

- In computer science, an AVL tree (named after inventors G.M. Adelson-Velsky and E.M. Landis) is a self-balancing binary search tree.
- In an AVL tree, the heights of the two child subtrees of any node differ by at most one; if at any time they differ by more than one, rebalancing is done to restore this property.
- The structure of an AVL tree is the same as that of a binary search tree but with a little difference. In its structure, it stores an additional variable called the Balance factor.
- Thus, every node has a balance factor associated with it.
- The balance factor of a node is calculated by subtracting the height of its right sub-tree from the height of its left sub-tree.
- Balance factor = Height (left sub-tree) – Height (right sub-tree)
- A binary search tree in which every node has a balance factor of –1, 0, or 1 is said to be height balanced. A node with any other balance factor is considered to be unbalanced and requires rebalancing of the tree.

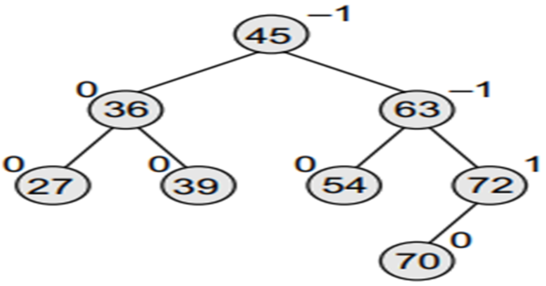

Figure 10.38 AVL tree

- LL rotation: The new node is inserted in the left sub-tree of the left sub-tree of the critical node.
- RR rotation: The new node is inserted in the right sub-tree of the right sub-tree of the critical node.
- LR rotation :The new node is inserted in the right sub-tree of the left sub-tree of the critical node.
- RL rotation: The new node is inserted in the left sub-tree of the right sub-tree of the

B. Operations on AVL Tree

Explain each of the following in details with suitable diagram.

- Traversing in an AVL Tree
- Searching for a Node in an AVL Tree
- Inserting a New Node in an AVL Tree
- Deleting a Node from an AVL Tree

7. Laboratory Exercise

A. Procedure

Write a C program to implement a AVL tree and show all the following operations in switch case,

i) Insertion
ii) Display Tree

B. Result/Observation/Program code:

Observe the output for the above code and print it.

8. Post-Experiments Exercise

A. Questions:

1. Explain with diagram LL,RR,LR,RR rotations of AVL Tree

2. Construct the AVL tree corresponding to given numbers 56,247,12,36,988,45,121,64,85,62.

B. Conclusion:

1. Summary of Experiment
2. Importance of Experiment

C. References:

1. S. K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011.
2. Reema Thareja; Data Structures using C; Oxford.