University of Colombo School of Computing
SCS 1308 - Foundations of Algorithms
Take-home

Instructions

- Try the following questions and upload your answer script as a zip file to the given link in the UGVLE on/before 02nd March at 6 pm.
- Note: Rename your zip file with your index number and name. (i.e: indexNo_Name.zip)

A financial technology (FinTech) company is developing a high-frequency trading system that must efficiently store and manage real-time stock transactions. Each transaction is identified by a unique transaction ID (TID) and must be quickly retrievable to ensure minimal latency in financial operations.

To achieve this, the company decides to use an AVL tree for storing transaction IDs, ensuring that insertion operations maintain a balanced search structure to keep lookup times optimal.

01. Write a C program that implements an AVL tree with the following functionalities

   A. Define the AVL tree node structure with attributes for key, height, left and right child pointers

   B. Implement AVL tree insertion, ensuring the tree remains balanced by applying the appropriate rotations when necessary.

   C. Implement functions for computing the balance factor and tree height.

   D. Implement a function to find the smallest and largest transaction ID stored in the AVL tree.

02. Modify an existing AVL tree implementation to count the number of rotations performed during insertions. Specifically, Count and classify rotations into four types: **LL (Left-Left), RR (Right-Right), LR (Left-Right), RL (Right-Left)**. Store counts in an array and display them after all insertions.

03. Test how different insertion orders affect the AVL tree's balancing behavior. Specifically, write a program that, Inserts transaction IDs into an AVL tree in three different orders:

Ascending order / Random order / Descending order

Tracks and prints the total number of rotations required for each insertion order and the final height of the AVL tree for each case. Compares results and explains why insertion order affects rotations.

04. The FinTech company generates thousands of transactions every second. Simulate the insertion of the following transaction IDs into the AVL tree in exact order: Consider this AVL tree.

$$\{2, 5, 4, 6, 7, 9, 8, 3, 1, 10\}$$

A. After each insertion, calculate the balance factor of the affected nodes.

B. Identify when and where rotations are triggered.

C. Classify each rotation as LL, RR, LR, or RL.

D. What is the final height of the AVL tree after inserting all transaction IDs?

E. If a standard unbalanced BST was used instead, what would be the expected worst-case height?

F. Compare the time complexity of search operations in AVL trees vs. BSTs.

G. How many total rotations (single and double) were performed during the insertions?

H. Which type of rotation (LL, RR, LR, RL) occurred most frequently? Why?

I. Does the order of insertion affect the number of rotations needed? Explain with examples.