DSA LAB 8

April 11, 2025

1 Submission Instructions

- The boilerplate code is provided in C++ for your convenience. You may choose to use it.
- The allowed programming languages are C and C++.
- You must submit a zip file named `Lab_8_RollNo.zip`, where `RollNo` is your roll number.
- The zip file should contain exactly two source files:
  - `Q1_RollNo.cpp` (Question 1 file)
  - `Q2_RollNo.cpp` (Question 2 file)
- You are not allowed to import any module except `iostream`.
- Question 1 is worth 15 marks, and Question 2 is worth 20 marks.
- Please try to submit both files as there will be partial marking if only one file is submitted.

2 Question 1: AVL Tree of Strings (Lexicographical Order)

Problem Statement: Implement an AVL tree that stores `string` keys such that the keys are maintained in lexicographical order.

Input (Hard Coded): Insert the following strings in the given order: `"delta"`, `"alpha"`, `"epsilon"`, `"beta"`, `"gamma"`.

Expected Output Format: Perform an in-order traversal that prints the strings in lexicographical order (e.g., `alpha beta delta epsilon gamma`).

3 Question 2: AVL Tree with Left Subtree Node Count

Problem Statement: Modify the AVL tree node structure to include an extra variable `leftCount` that stores the number of nodes in the left subtree. Update the insertion and rotation functions so that `leftCount` remains accurate.

Input (Hard Coded): Insert the following keys in the given order: 50, 30, 70, 20, 10, 40, 60, 80.

Expected Output Format: Perform an in-order traversal that prints each node's key along with its left subtree count. For example, if the node with key 50 has three nodes in its left subtree, it should be printed as `50:3`.