COMP - 4433 WA

Assignment 3

Due Date: March 09, 2024 @ 23:55

Total Marks: 100

AVL Tree Implementation and Operations

Introduction:

An AVL tree is a self-balancing binary search tree. It maintains a balanced tree structure by performing rotations whenever necessary during insertion and deletion operations. In this programming assignment, you will implement an AVL tree and various operations on it.

Assignment Objectives:

- Implement the AVL tree data structure.
- Implement basic operations such as insertion, deletion, search, and traversal.
- Ensure that the tree remains balanced after each operation.
- Test the implemented operations thoroughly.

Assignment Tasks:

AVL Tree Implementation:

- Define a class for AVL tree nodes.
- Implement a class for AVL trees with methods for insertion, deletion, search, and other necessary operations.
- Ensure that the tree remains balanced by implementing rotation methods (left rotation, right rotation, left-right rotation, right-left rotation) as required.

Basic Operations [40 marks]:

a. Insertion:

Implement a method to insert a new node into the AVL tree while maintaining the balance factor.

b. Deletion:

Implement a method to delete a node from the AVL tree while ensuring that the tree remains balanced.

c. Search:

Implement a method to search for a given key in the AVL tree.

d. Traversal:

Implement methods for in-order, pre-order, and post-order traversal of the AVL tree.

Balancing Operations [20 marks]: A rotation.cpp file is provided which has the template of rotation operations to balance a tree. You are required to call the avl_tree.cpp file as header and rotation will be performed when the tree becomes unbalanced by inserting or deleting an element as specified in test-avl.cpp file.

Testing [30 marks]:

- Develop a set of test cases to validate the correctness and efficiency of the implemented AVL tree operations.
- Align your AVL implementation as specified in the test-avl.cpp and add more test cases to prove the accuracy of your implementation.
- Test edge cases such as insertion of duplicate keys, deletion of non-existent keys, etc.

Deliverables:

- Source code files containing the implementation of the AVL tree and its operations.
- Create a README file to explain how your test case should be executed to produce the specified output.
- Test cases and their results demonstrating the correctness and efficiency of the AVL tree operations.

Submission Guidelines [10 marks]:

- Submit all source code files, documentation and test cases, in a well-organized manner.
- Ensure that the code is properly commented and follows best coding practices.
- Provide clear instructions on how to compile and execute the code.
- Copy all files in a folder and give it a name 4433-WA-A3-[STUDENT-FIRST-NAME] compress it and upload it on D2L by the specified deadline .
- Submit the assignment by the specified deadline.

- **Not adherence** to the submission guidelines will result in marks deduction by 20%.

Late Penalty

The late penalty for assignment is [2.5^i] (2.5 to the i-th power, rounded to the nearest integer), where i>0 is the number of days you are late. So, if you hand-in your assignment 1 day late, you will be penalized 3%, a delay of 2 days will decrease your grade by 6%, 3 days is penalized 16% and 4 days takes 39% off your grade. You cannot be more than 4 days late, Extensions will be granted only by the course instructor. If you have serious medical or compassionate grounds for an extension, you should take supporting documentation to the office of the Dean of your faculty, who will contact the instructor.

Important Note:

- Plagiarism will lead to serious consequences and a grade of zero for the assignment. The similarity check will be used to detect possible cheating cases where both students will be awarded **ZERO** if found guilty.
- No support from Chat GPT or any other source is allowed and, **IF** will be detected by in place sophisticated softwares **THEN** will be dealt with by University prescribed rules.
- Feel free to seek assistance from the instructor or teaching assistants if you encounter any difficulties during the assignment.

****************** **END of ASSIGNMENT 3** ****************