جــــامعة نيـويـورك أبــوظبي

NEW YORK UNIVERSITY
ABU DHABI

DATA STRUCTURES

Lab Exercise 12 : AVL Tree

In this lab, you will extend the Binary Search Tree (BST) class you implemented in one of the previous labs to support *AVL Tree* functionality. AVL trees are self-balancing binary search trees, where the difference between the heights of left and right subtrees (i.e., the *balance factor*) is at most 1 for every node. When the balance factor condition is violated due to insertions or deletions, the tree must perform *rotations* to restore balance.

You will add the following methods to your BST class to convert it into an AVL tree:

1. `int` **`balanceFactor`**`(Node* ptr)` (2 Point)
   - Calculate and return the balance factor of a given node.
   - You need to use the method **`height(Node* ptr)`** defined in the BST lab.
   - For a nullptr/empty node, the height is defined as -1.

2. `void` **`rotateRight`**`(Node* z)` (2 Point)
   Perform a right rotation around node z.
3. `void` **`rotateLeft`**`(Node* z)` (2 Point)
   Perform a left rotation around node z.
4. `void` **`rebalance`**`(Node* ptr)` (3 Point)
   A method that rebalances a tree/subtree rooted at node ptr. After every insertion or deletion, traverse up to the root from the affected node, and if the node becomes unbalanced, perform the appropriate rotation to balance it.

5. **Comments:** (1 Point)
   Use meaningful comments to explain the logic behind your implementation, especially around tricky sections like rotation and balance factor calculation.
   Proper indentation and naming conventions are expected.

CODE OF CONDUCT

All assignments are graded, meaning we expect you to adhere to the academic integrity standards of NYU Abu Dhabi. To avoid any confusion regarding this, we will briefly state what is and isn't allowed when working on an assignment/lab-task.

Any documents and program code that you submit must be fully written by yourself. You can, of course, discuss your ideas with fellow students, as long as these discussions are restricted to general solution techniques. Put differently, these discussions should not be about concrete code you are writing, nor about specific results you wish to submit. When discussing an assignment with others, this should never lead to you possessing the complete or partial solution of others, regardless of whether the solution is in paper or digital form, and independent of who made the solution, meaning you are also not allowed to possess solutions by someone from a different year or course, by someone from another university, or code from the Internet, etc. This also implies that there is never a valid reason to share your code with fellow students, and that there is no valid reason to publish your code online in any form.

Every student is responsible for the work they submit. If there is any doubt during the grading about whether a student created the assignment themselves (e.g. if the solution matches that of others), we reserve the option to let the student explain why this is the case. In case doubts remain, or we decide to directly escalate the issue, the suspected violations will be reported to the academic administration according to the policies of NYU Abu Dhabi.

https://students.nyuad.nyu.edu/campus-life/community-standards/policies/academic-integrity/