GTU Department of Computer Engineering

CSE 222/505 - Spring 2022

Homework #07 Report

Şiyar Tarık Özcaner

20010400421

1. System requirements

In this assignment there is 3 questions, first one asks for a method that will create a BST by compiling together a binary tree and an array of inputs. Second one asks for a method that will convert a regular BST to an AVL tree. And finally last question it is asked to implement a custom *skiplist* with a modified insertion algorithm that by a calculated chance inserts the input to the higher levels too.

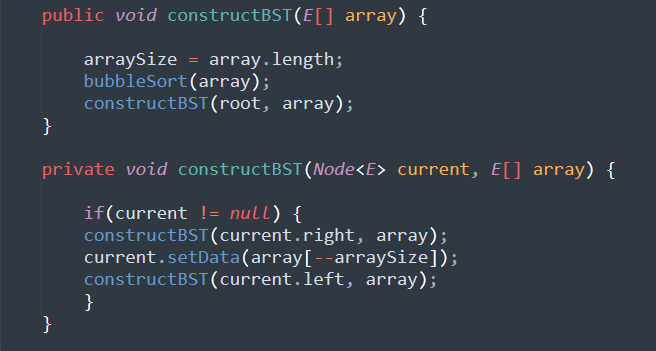
1. Class diagrams

* Is in the directory

1. Problem solutions approach

Q1:

Well implementation for a binary tree and BST and AVL tree is needed and the lecture book in this regard is used as a guide, after implementations are in place method named “constructBST” is implemented in the binary tree class. This method modifies the binary tree as such that final form of the binary tree is as it is a BST.

Theoretical Time Complexity: 

Here sorting algorithm “bubble sort” is used and main point of complexity is that which is *O(N2)*

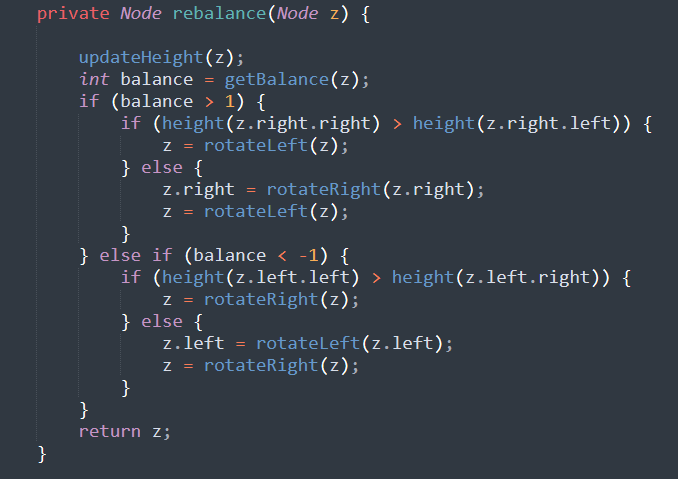
After that there is N times recursive call *O(N)* which is not a concern because of *O(N2)*

Q2:

All of the sorting algorithms are pretty standard, last one being binary search.

Theoretical Time Complexity:

After all elements of BST extracted by traversing which is an operation of O(N) since all the nodes are traversed



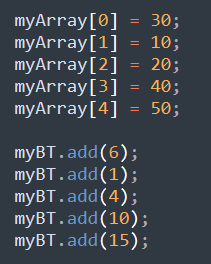
“Insert” method which is a very standard add method of a BST other than the balancing act calls this “rebalance” method, if tree is balanced than except couple of comparisons (“updateHeight” O(1) since height is stored, “getBalance” O(1) since stored height are operated on) no operation is done so O(1), if tree is unbalanced then couple of basic rotation operations are made to fix that. So overall time complexity of the operation can be considered as O(N) because of “getArray” method.

Q3:

**Is not solved**

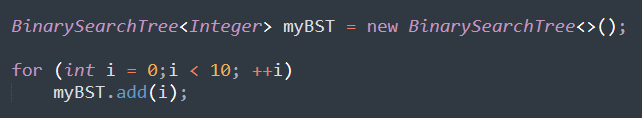
1. Test cases

For Q1 An array and BT is created with the same number of nodes



then elements of array are distributed by “constructBST” method

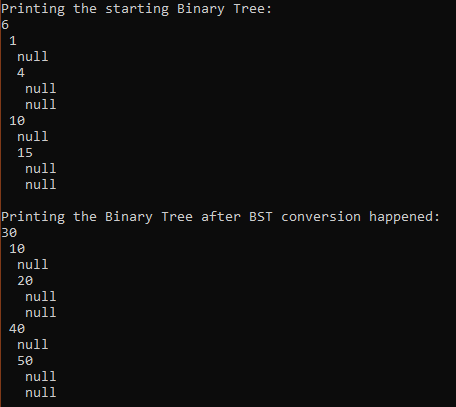
for Q2 an specifically unbalanced BST is created by the following for loop



And then “BSTtoAVL” method extracts elements of BST and constructs a balanced AVL tree.

1. Running command and results

After a simple “make” command in the terminal these are the result:



For Q2

