Programming Assignment 7

Due 6 NOV @ 11:59pm

Modify the provided binary search tree implementation, named AVLTreeST, and make it balanced using AVL rotations. The current implementation is a BST and does not store nodes in a balanced manner. A template is provided that will read in and execute a sequence of operations. The template will also execute a performance test (note: an analysis is NOT required for this assignment).

The Node class with height instance variable has been provided. Make sure you can properly update the height information before attempting the rotations. Leaf nodes (i.e. left and right links are null) should have height of 0. Utility methods max(...) and height(...) methods are provided and optionally used. The height method considers a null link height to be -1.

The balance method checks to ensure the height information is correct and that the tree is AVL balanced. The current implementation produces an unbalanced tree.

Grading Notes

You must:

- Use the template provided for you
- Have a style (indentation, good variable names, etc.)
- Comment your code well (no need to over do it, just do it well)

You may not:

- Make your program part of a package.
- Use *code* from anywhere except your own brain.

Submission Instructions:

- Name a folder with your gmu username
- Put your java files in the folder (but not your .class)
- Zip the folder (not just the files) and name the zip “username-pa2.zip”
- Submit to blackboard

Grading Rubric

No Credit:

- Non-submitted assignments
- Late assignments
- Non-compiling assignments
- Non-independent work

1pt	Submission Format
1pt	Style and Comments
2pt	put case 1 and 2
2pt	put case 3 and 4
1pt	rotateRightChild method
1pt	rotateLeftChild method
1pt	rotateDoubleRight method
1pt	rotateDoubleRight method

Example Run

```
>java AVLTreeST operations.txt
size=9
Final symbol table=[2->student2, 3->student3, 4->student4, 6->student6, 10->student10, 15->student15, 18->student18, 19->student19,
                    20->student20]
Balanced? true
```

Example Performance Run

```
> java AVLTreeST 500000
Put 500000 items took 686.331486ms
Balanced? true

> java AVLTreeST 1000000
Put 1000000 items took 957.238049ms
Balanced? true

> java AVLTreeST 2000000
Put 2000000 items took 2659.711406ms
Balanced? true

> java AVLTreeST 4000000
Put 4000000 items took 4756.338839ms
Balanced? true
```