CIS 303 Algorithm Analysis and Design
Assignment 5b: AVL Tree
Program and Analysis

L. Grabowski

April 4, 2020

Directions:

- This is an individual programming and analysis assignment. 60 points possible.
- The analysis must be typeset in LaTeX. See the template available in Moodle to get started.
- Carefully read and follow the instructions for the program and the analysis and report, below. Please note that the grading rubric for the assignment will be available in Moodle.

1. (30 pts) **Program.** For this problem, you will implement selected methods for AVL trees.

   There are three source files in the `start.zip` folder:

   - `AVLClient.java`: simple skeleton of a client program for BSTs and AVL trees. You will need to add to this client to build and test your AVL tree. Make sure that you test your tree thoroughly for correctness, providing the program with cases that will test all cases for balancing. You will hand in this first version of your client code, NOT the version you use for experiments.
   - `BSTNode.java`: binary search tree node class. This class will be used in your experiments for Part 2. Note that this implementation (and the `AVLNode` implementation that is based on it) will not put duplicate values in the tree. Your test client for Part 1 and experiment code for Part 2 must account for this when generating data for the trees.
   - `AVLNode.java`: partial implementation of the `AVLNode` class. The tree is simply represented by the overall root `AVLNode`. Right now, when integer data are inserted at the root, a regular BST tree is built; that is, there is no balancing. You will write the implementation for the `balance` method and the four rotation methods to make the AVL tree work.

2. (30 pts) **Analysis and Report.** Refer to the "Sample Lab Report" for information and guidance about writing your analysis.

   (a) (10 pts) **Introduction and hypothesis.** Based on the implementations of `BSTNode` and `AVLNode`, develop a hypothesis about the performance of the two types of trees **in the worst case**. What will be worst case be for this problem? Remember that a hypothesis must be falsifiable and based on measurements that will come from your experiments.

   (b) (20 pts) **Experiments, analysis, and discussion.**

      i. **Experiments.** Use a modified version of `AVLClient.java`, collect performance data for `BSTNode` and `AVLNode` for a minimum of the following tree sizes (*i.e.,* number of nodes in the tree): $\{100, 500, 1000, 2500, 5000, 7500, 10000, 15000, 20000, 25000\}$ (see NOTE below, however). If the system you are running your experiments can handle it, include $N = 50000$. For each size of N, be sure to test the two different tree types with the same data set. You may generate the data for your trees in any way you wish and you may use additional tree sizes in your experiments. Remember that you must arrange your data in **the worst case**.

NOTE: I recommend that you connect to the lab server to run these experiments. If you run on your local machine, you may not be able to run experiments for the array sizes I'm asking for in these experiments. There will be variations between machines.

Keep in mind that you will need to add to your client code to conduct your experiments, which involve testing the performance of your AVL tree against a BST that has no balancing. ONLY the client code should be altered for experiments. You must not change the `AVLNode` or `BSTNode` classes for testing (they must still work with my client code when you submit your assignment). You are free to write helper methods for the client program to support whatever you need in testing and experiments.

ii. **Analysis.**

- A. Create a plot showing the data for both the BST and AVL tree for each size of N (*i.e.,* you will 2 lines on your plot, one for BST and one for AVL tree).
- B. Note that the curves will be pretty jagged if you use only the required values of N. You will only have 10 data points to draw each curve.

iii. **Discussion.**

- A. Discuss your experiment results, what they show, and how they either support or refute your hypothesis.
- B. Be sure to include your plot in your lab report. To do that, you need to convert your plot to a graphics format (.jpg, .png, or .pdf) and include it as an image in your report. If you're using Excel, you can easily right click on the plot and export it as a picture. I have put the necessary LaTeX code to insert an image in the lab report template (currently commented out). You can move the code to the appropriate spot in your document.
- C. Suggest follow-up experiments that may provide additional insight or help better explain the data that your experiments produced.

Assignment Submission. You will turn in the following files for your assignment. All files will be submitted only in electronic form (no hard copies needed). You may place all your files in a folder and zip them so you only submit the zipped file. Any compression/archive format should be fine; if there's a problem with a particular format or compressed file, I will let you know.

1. `AVLClient.java` file (pre-experiment version). I will test your code for correctness with my own client, but I want to see how you test your AVL tree (this will be part of your grade).

2. `AVLNode.java` file.

3. The data file/files that was/were produced by your tests (*i.e.,* your output files).

4. The data spreadsheet that you produced, including both the raw data and the plot you generated. If you used a different tool to produce your data plot (such as a Python script), you may turn that in instead of the spreadsheet.

5. Analysis lab report (part 2), typeset in LaTeX. Be sure to edit the author and assignment information, as shown in the template. The date will be automatically generated by LaTeX. Turn in the PDF of the file, not the `.tex` source. Please do not hesitate to come see me in my office if you need LaTeX help.

Submit all required files to the Moodle assignment by the published deadline.