CS 124 Project 3

Due on Blackboard by Wednesday, March 7

For this project, you will perform a series of searches for 1000+ objects stored in a Binary Search Tree, an AVL Tree, and a Splay Tree. For each search, you will record how many objects you had to visit to complete the search. You will analyze your results from the different data structures.

Implement

- You should have your 1000+ objects stored in a vector (I recommend you start with your Project 2 code). Add at least three more dummy objects to the vector. That way, you only need integer indexes to determine which objects you are searching for, and you will have at least three objects in the vector that will not be in the tree structures.
- Say you have N objects in your vector. You will need to create three files of integers: a sorted list in range [0, N-1]; a random ordering of the same integers; and a list of integers [0, N/5] that repeat each integer five times in a row (i.e. 0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 etc.)
- Store the objects (excluding the dummy objects you added to the vector) in a Binary Search Tree, an AVL Tree, and a Splay Tree. I recommend that you start with the textbook code (http://users.cs.fiu.edu/~weiss/dsaa_c++4/code/). Note that you will need to overload the <, >, ==, and << operators to compare two objects of your class. You should copy the entire class from the textbook (don't cherry pick parts or your program will have memory leaks). Any code that is not original **must be cited in your writeup** (including textbook code).
- Modify the code so that a search of the tree also stores the depth of the last node visited. I recommend that you pass the address of an integer to the contains method and modify it inside the method(s).
- Read in from each of the three files, an integer at a time, searching for the object at that integer index in the vector. Search the BST, the AVL tree, and the Splay tree. Store the depths of each search.
- Analyze the data. Plot graphs of the depths for the 1000+ searches in each of the three tree structures. Look at maximums, minimums, averages, and patterns, compare and contrast the different tree structures for each data file, and draw conclusions about when you would use each tree structure. Discuss complexities and their effects. All of this will go in your writeup.
- You must submit your .cpp file(s), your data file(s) (including the three integer files), and your writeup. Please submit your writeup in PDF format.

Extra Credit

To earn up to 10 extra credit points (at the grader's discretion), you can get more thorough results. This can include:

- Setting timers to record how long it takes you to search for the objects in each data structure
- Creating more integer files and seeing their effects
- Performing the same experiments on 100, 200, 300, …N objects and graphing the results

Grading

The project is out of 80 points.

5 pts	Program compiles and runs.
5 pts	Code style. Readable, naming style is consistent, comments where appropriate.
5 pts	Created three files of integers according to the specifications.
15 pts	Storing the data in a BST, AVL tree, and Splay tree.
30 pts	Searching for objects from the files in the three trees, recording the depths reached.
15 pts	Analyzing the results.
5 pts	Writeup: professional, grammatically correct, cites sources.