CS 3410 - Homework 04

Due date: see course Schedule and Blackboard.

Overview

You will compare how long it takes to build a BinarySearchTree and a Red-Black tree using random integers and patterned integers.

Requirements

1. Obtain the download, *hw4.zip*.

2. Open *TestingShell.java* and run. It will take 30 minutes to an hour.

3. The output is in *ExperimentResults.txt* and consists of four sets of data. The first set of data can be described like this: Use a Red-Black Tree to time how long it takes to insert 100,000 *patterned* integers. Do this 10 times and take the average. This is the value 0.103643 shown in the example below. Next, repeat for 200,000 inserts; ...; up to 1,000,000 inserts.

Data Structure-Data	n	Time (sec)
Red-Black-Patterned	100000	0.103643
Red-Black-Patterned	200000	...
Red-Black-Patterned	...	
Red-Black-Patterned	1000000	...

   The second set of data is exactly the same except the Binary Search Tree was used. The third and fourth sets use the Red-Black Tree and the Binary Search Tree, respectively; however, this time, inserting random integers.

4. Object: find out what the "pattern" of numbers is that are being inserted for the patterned runs. You can figure it out by looking at the *for* loop below and working the numbers out (*numVals* is 100,000 the first time, then 200,000, etc). Or, you can put some print statements in one of the patterned runs to see what the pattern of numbers is. Look for the first occurrence of this line:

   ```
   for( int val = GAP; val != 0; val = ( val + GAP ) % numVals )
   {
   ```

   and print *val* out just inside the loop. You'll have to stop the program quick to see the numbers/pattern.

5. (15 points) Start a write-up for this assignment. Describe the pattern of numbers.

6. (35 points) Make a graph of both (Red-Black and Binary Search Tree) sets of patterned data (n on the x-axis and time on the y-axis). Describe what the graph shows and explain your results with your knowledge of theory behind the data structures.

7. (35 points) Make a graph of both sets of random data. Describe what the graph shows and explain your results with your knowledge of theory behind the data structures.

8. (20 points) Compute the average number of single rotations required per insert when 100,000 patterned values are inserted. In other words, count how many single rotations are required and then divide by 100,000. Repeat this 10 times and average the results. Repeat for double rotations. Repeat for 200,000; ... Repeat for random inserts. Make a scatter graph with 4 lines (Single Rotations-Patterned, Double Rotations-Patterned, Single Rotations-Random, Double Rotations-Random). If two graphs works better, do that instead. Describe what you see in the graphs. Is there evidence that the values rise as the number of inserts increases?

   All you need to do is put in counters to count single and double rotations. To do this, you'll need to study the *RedBlackTree.java* code carefully. Obviously, you'll need to be careful to reset the counters at the right time, and to pull the current values and average them at the right time.

   Note: the zip file also contains code for these data structures: Avl Tree, AA Tree, Splay Tree. You are welcome to explore them!

Deliverables

1. A Word document in this format:
   a. Title page:

      CS 3410 – HW 04
      Name
      Date

   b. Answers to 5,6,7,8 above. Include the graphs in the text at the appropriate places.

2. Your Excel spreadsheet used to make the graphs.