Hyrum Johnson 10/26/2016

Analysis Assignment 8

1. Who is your programming partner? Which of you submitted the source code of your program?

   A) **Again I programmed this assignment by myself due to the fact that we are a bunch of anti social nerds and some of us would rather do extra work then have a conversation with someone we don't know. It was a surprise that i found a partner in the first couple of assignments.**

2. Evaluate your programming partner. Do you plan to work with this person again?

   A) **I think I did pretty good. I will make sure and find a partner for the next assignment.**

3. Design and conduct an experiment to illustrate the effect of building an N-item BST by inserting the N items in sorted order versus inserting the N items in a random order. Carefully describe your experiment, so that anyone reading this document could replicate your results. Submit any code required to conduct your experiment with the rest of your program and make sure that the code is well-commented. Plot the results of your experiment. Since the organization of your plot(s) is not specified here, the labels and titles of your plots(s), as well as, your interpretation of the plots is critical.

□One suggestion for your experiments is:

- Add N items to a BST in sorted order, then record the time required to invoke the contains method for each item in the BST.

- Add the same N items to a new BST in a random order, then record the time required to invoke the contains method for each item in the new BST. (Due to the randomness of this step, you may want to perform it several times and record the average running time required.)

- Let one line of the plot be the running times found in #1 for each N in the range [1000, 10000] stepping by 100. (Feel free to change the range, as needed, to complement your machine.) Let the other line of the plot be the running times found in #2 for each N in the same range.

   A) **Check all graphs. Red line is in nanoseconds where blue line is data set. As you can see as the data set gets bigger the time to run takes longer in each case.**

4. Design and conduct an experiment to illustrate the differing performance in a BST with a balance requirement and a BST that is allowed to be unbalanced. Use Java's TreeSet (http://docs.oracle.com/javase/7/docs/api/java/util/TreeSet.html) as an example of the former and your BinarySearchTree as an example of the latter. Java's TreeSet is an implementation of a BST which automatically re-balances itself when necessary. Your BinarySearchTree class is not required to do this. Carefully describe your experiment, so that anyone reading this document could replicate your results. Submit any code required to conduct your experiment

with the rest of your program and make sure that the code is well-commented. Plot the results of your experiment. Since the organization of your plot(s) is not specified here, the labels and titles of your plots(s), as well as, your interpretation of the plots is critical.
□One suggestion for your experiments is:

- Add N items to a TreeSet in a random order and record the time required to do this.

- Record the time required to invoke the contains method for each item in the TreeSet.

- Add the same N items (in the same random order) as in #1 to a BinarySearchTree and record the time required to do this.

- Record the time required to invoke the contains method for each item in the BinarySearchTree.

- Let one line of the plot be the running times found in #1 for each N in the range [1000, 10000] stepping by 100. (Feel free to change the range, as needed, to complement your machine.) Let the other line of the plot be the running times found in the #3 for each N in the same range as above.

- Let one line of a new plot be the running times found in #2 for each N in the same range as above. Let the other line of plot be the running times found in #4 for each N in the same range.

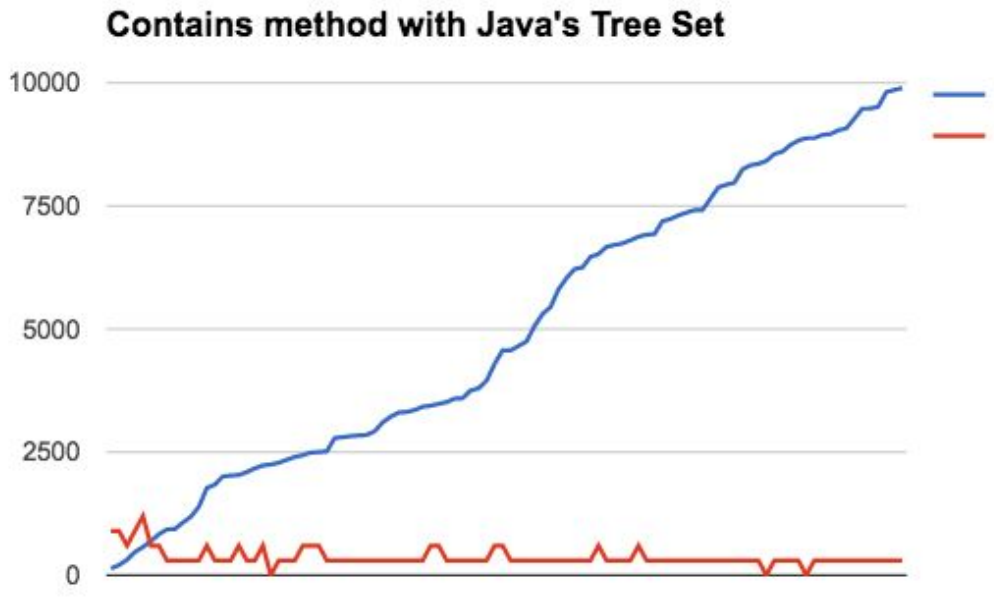
A) **Check all graphs. Red line is in nanoseconds where blue line is data set. As you can see as the data set gets bigger the time to run takes longer in each case.**

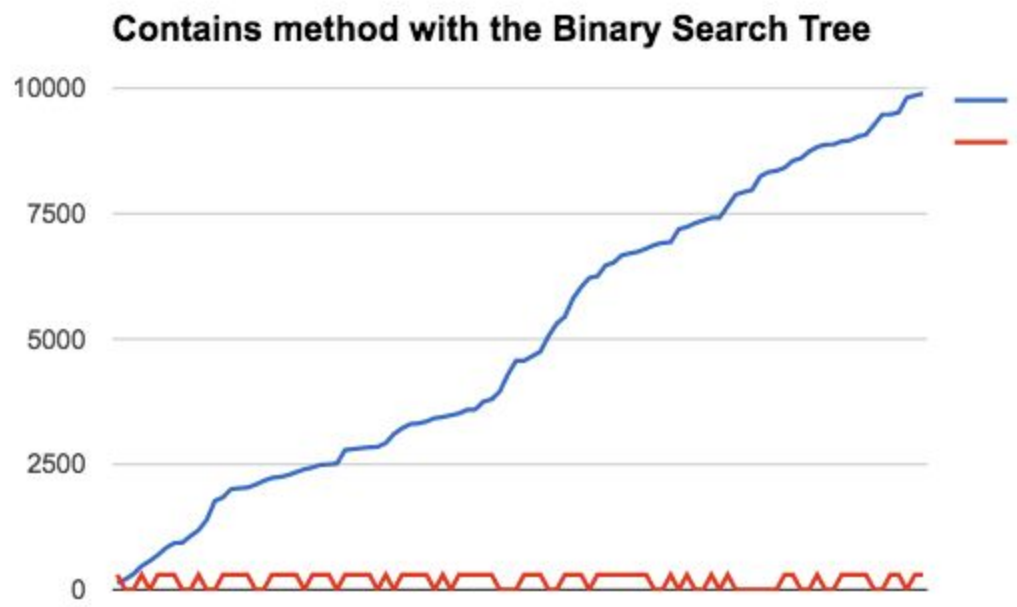
5. Many dictionaries are in alphabetical order. What problem will it create for a dictionary BST if it is constructed by inserting words in alphabetical order? Explain what you could do to fix the problem.

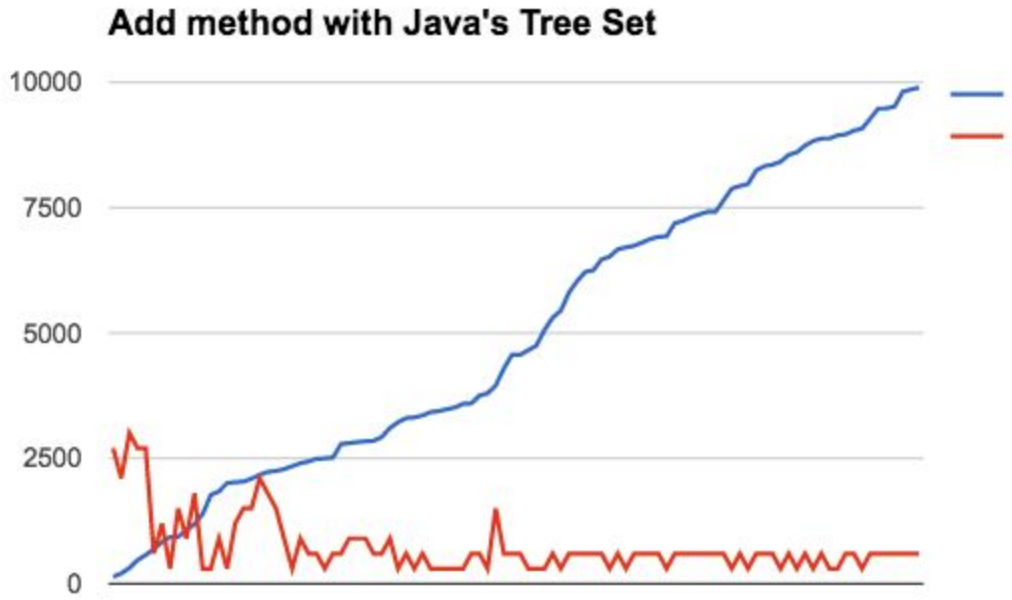
A) **The problem with this is it will require a lot of space and time complexity especially when the height of the tree is big. A better thing to do is to find the closest set of words that can be defined as the number of characters away a specific word is from the chosen word there are many different types of algorithms you can use to do this.**

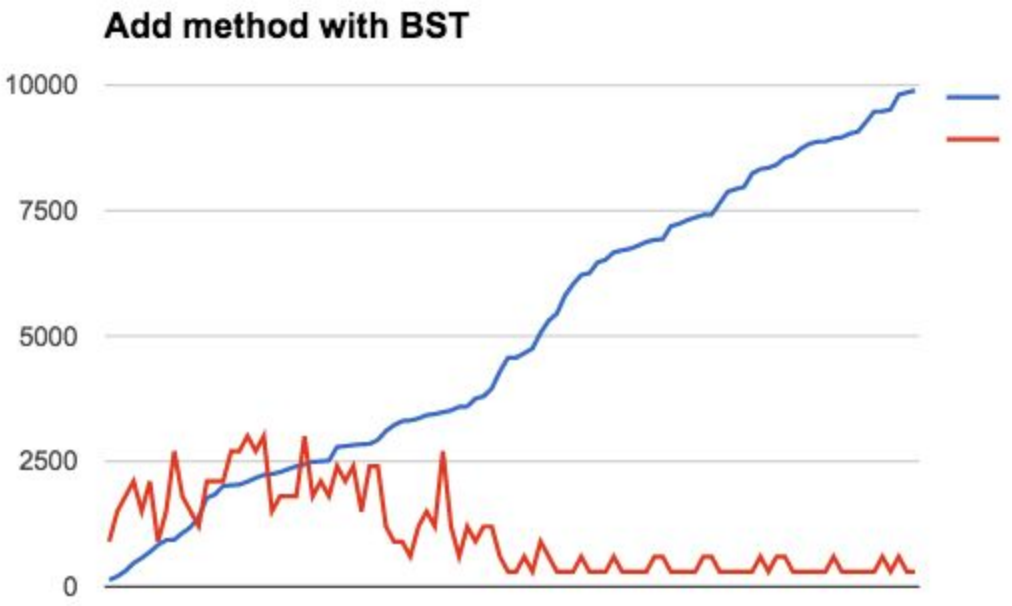
6. How many hours did you spend on this assignment?

A) **15 - 17 I usually have to spend a little more time than average.**


Contains method with Java's Tree Set
10000
7500
5000
2500
0


Contains method with the Binary Search Tree
10000
7500
5000
2500
0


Add method with Java's Tree Set
10000
7500
5000
2500
0


Add method with BST
10000
7500
5000
2500
0