D Ben Green
CS 2420
Feb 5, 2015

Analysis Assignment3

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

My partner is Xavier Humberg. I submitted the code for assignment 3.

2. How often did you and your programming partner switch roles? Would you have preferred to switch less/more often? Why or why not?

We switched a few times. Overall he was mostly the driver and I was mostly the navigator. I was fine with the times we switched because it was a nice change of pace, but we still were in roles that fit us most.

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

My partner has a pretty good knowledge of programming like I do. It is kind of hard to work with him at times because he is a very different person than me. Also, he sometimes is competitive and doesn't want to admit when he is wrong.

4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development time.)

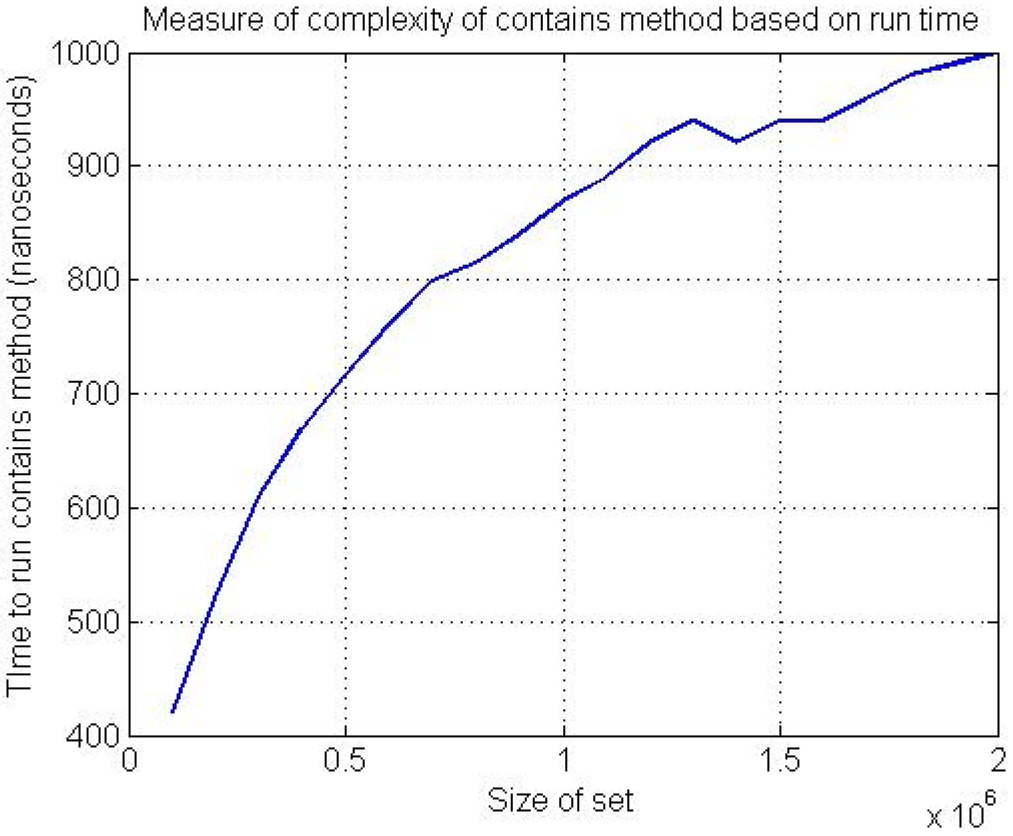
We would have to write code for more functions. The List would probably end up being more efficient in terms of run time, but would take more time to develop because of the extra methods.

5. What do you expect the Big-O behavior of MySortedSet's contains method to be and why?

I expect it to be O(NlogN). The logN comes from the binary search. The N comes from the comparing.

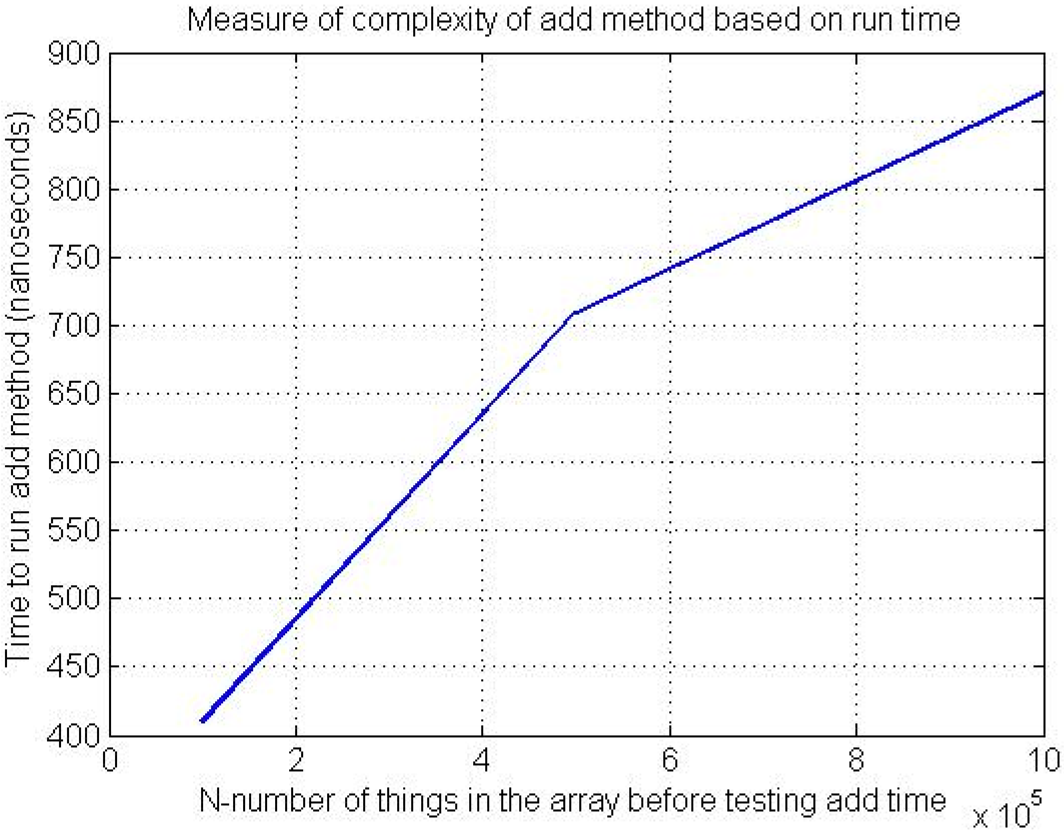
6. Plot the running time of MySortedSet's contains method for sets of sizes 100000 to 2000000 by steps of 100000. Use the timing techniques demonstrated in Lab 1. Be sure to choose a large enough value of timesToLoop to get a reasonable average of running times. Include your plot in your analysis document. Does the growth rate of these running times match the Big-oh behavior you predicted in question 5?

The growth of these running times does not match O(NlogN).



7. Consider your add method. For an element not already contained in the set, how long does it take to locate the correct position at which to insert the element? Create a plot of running times. Pay close attention to the problem size for which you are collecting running times. Beware that if you simply add N items, the size of the sorted set is always changing. A good strategy is to fill a sorted set with N items and time how long it takes to add one additional item. To do this repeatedly (i.e., timesToLoop), remove the item and add it again, being careful not to include the time required to call remove() in your total. In the worst-case, how much time does it take to locate the position to add an element (give your answer using Big-oh)?

The complexity of the add method starts with the binary search: logN. The comparing should be N. We also have another comparison which is irrelevant to the leading term. The add method should be O(NlogN).



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

We spent about 6.5 hours on this assignment.

I spent about 1.5 hours on this analysis document (didn't expect this work based on the last analysis).