John Chambers Analysis doc

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

Seth Kingston

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

Pretty often. We usually had one of us typing and the other navigating, and then we'd switch whenever one of us had an idea or came to a conclusion of how to do the methods.

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

Yes. I do plan on working with Seth again. We work together well and can come up with good solutions.

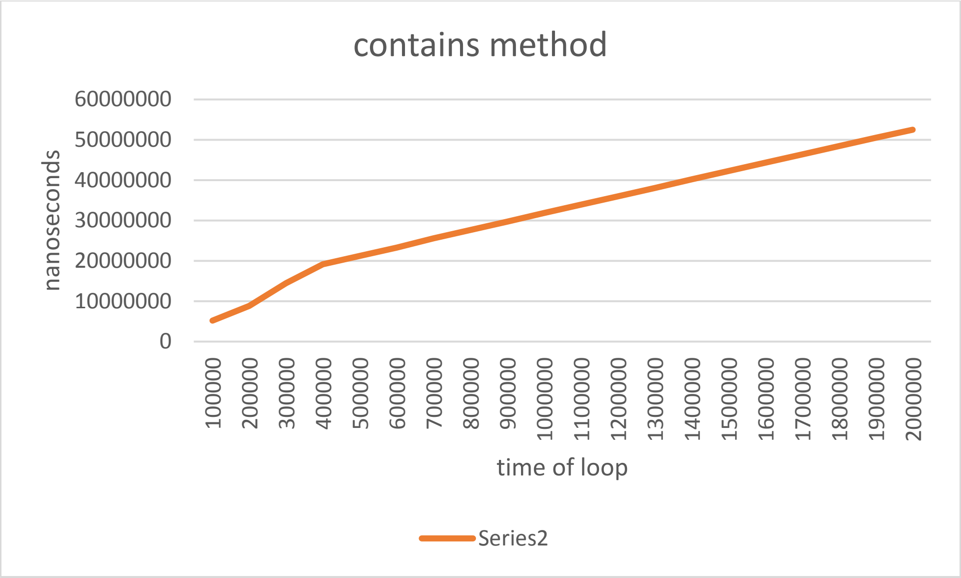
4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development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

It seems like it'd be more efficient because java lists are organized in such a way that it's be easier to iterate through them and you could change the size of them. The development would have probably been cleaner, and the running time would have become much more efficient.

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

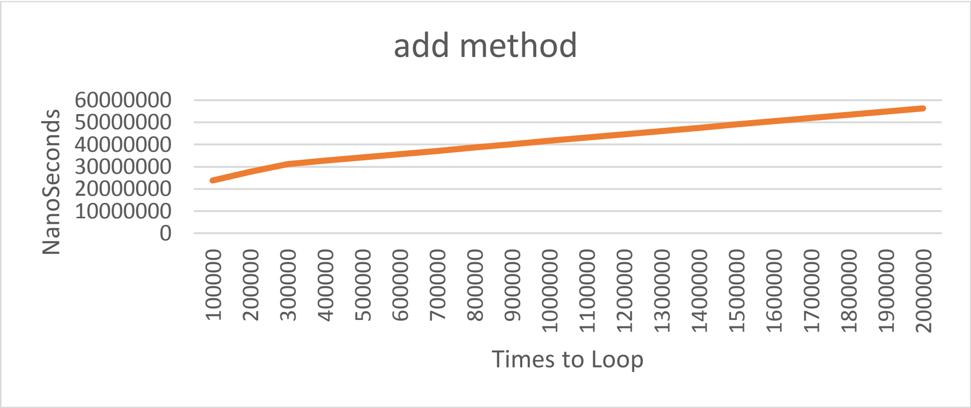
log N. because I feel like it would be much easier for it to search the method when it was small, but get increasing larger as N went on.

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?



A little bit. It is off though, and it looks more like a Linear graph or perhaps an N logN curve.

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)?



in the worst case, it would take N log N time to locate the position of an element.

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

About 26 hours.