Leonardo Roese

When you are satisfied that your program is correct, write a brief analysis document. The analysis document is 30% of your Assignment 3 grade. Ensure that your analysis document addresses the following.

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

   Kayden Thomson, I turned in the Source Code

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

Every 20 minutes, no because I believe that 20 minutes is enough time to keep both parties focused. It is a lot easier to get distracted as the navigator then the driver.

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

Great. He was there every time and the whole time we needed to work together, yes I would work with him again.

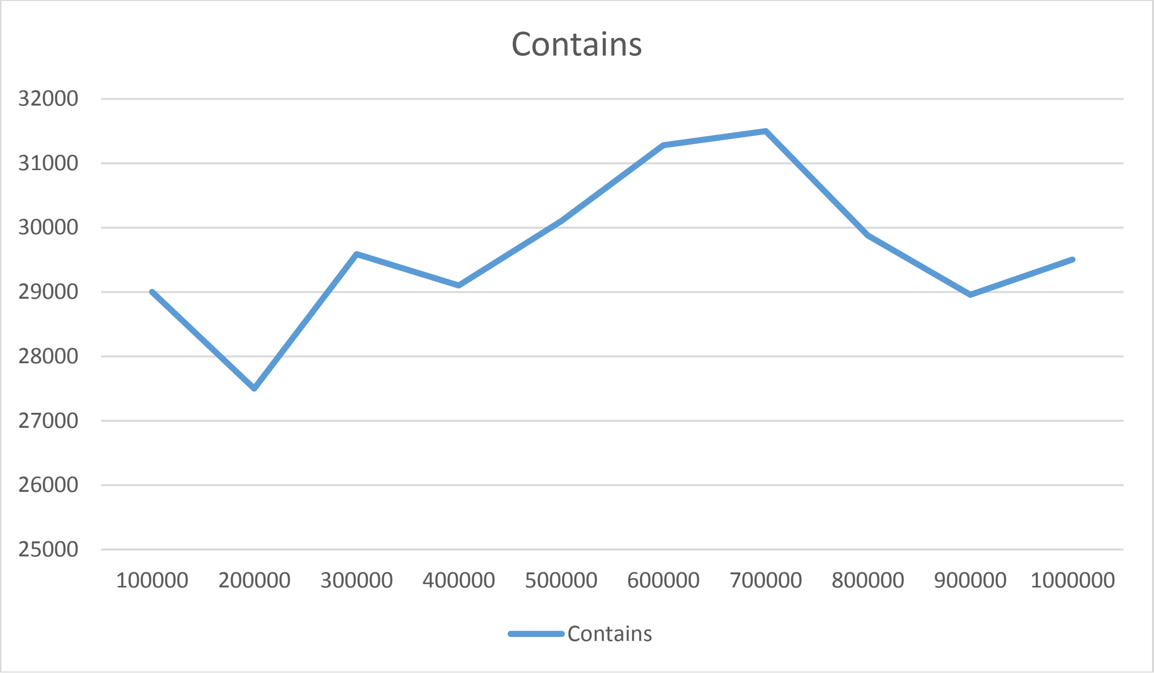
4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development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

I believe that it would have been less efficient to use a java list instead of a basic array. In our add method we are adding in a sorted manner, if we had used the Array list we would have to add the objects then sort them after. Time complexity of our add and contains method at worst is O(log N) while for a java list at worst for add and contains is O(N). Programming time however would have been able to implement faster using a java list since it has built in methods to help us with sorting and iterating.

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

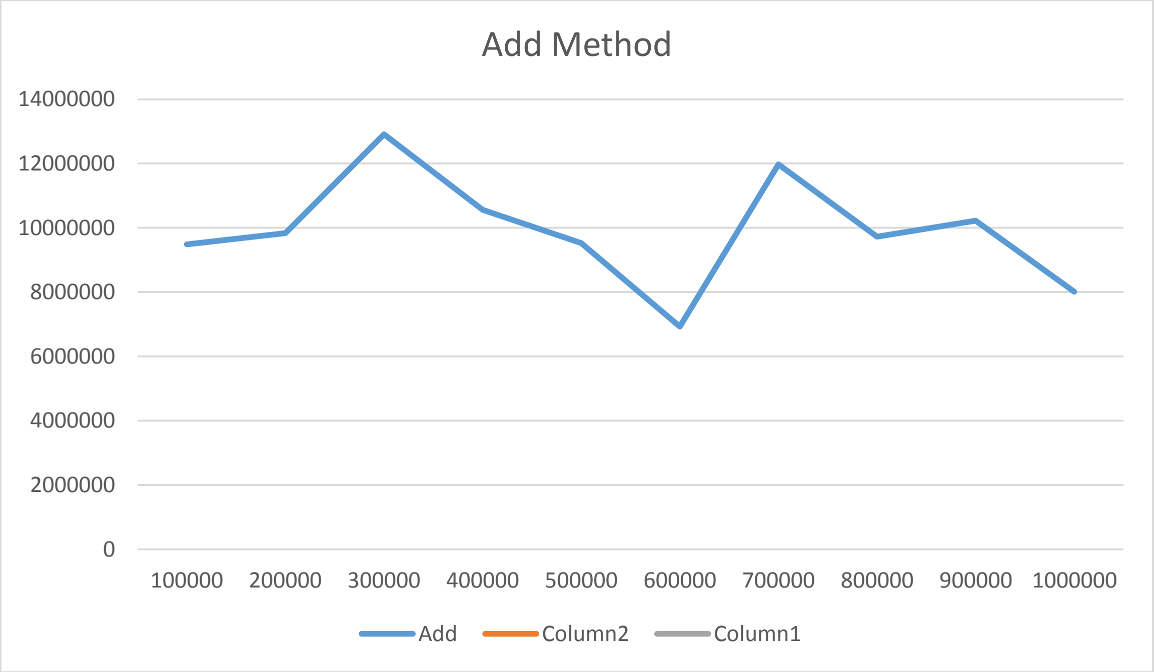
I believe it is O(log N) due to the fact that we use a binary search in order to see whether the array contains the object.

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 graph that I got is a bit more erratic than I thought it was going to be. I thought it was going to represent the O(Log N) graph.

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)?



It takes longer than contains due to the fact that after finding the location you also need to shift all the elements to the right then add the object. When you have an array of 100000 objects shifting everything to the right takes some time.

Worst case takes O(N) because of the need to shift the elements to the correct position.

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

14 hours