Stacey Kirby
Assignment 3 Analysis

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

My programming partners name is Neeka Ebrahimi. My partner Neeka Ebrahimi is submitting the 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?

We switched off every hour, but sometimes with shorter intervals. I would have liked if we had switched more often, but at the same time I enjoyed how we would each have to do separate functions, and one person could volunteer ideas to be used. Another reason why I liked switching less often was that it allowed the person typing to be able to insert all of their ideas, and if need be, have the other person help them with problems that arise. It also had allowed me to get into the critical thinking mode where I thought of potential problems that could have arisen or minor technical mistakes that the compiler would not have caught.

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

I highly enjoyed working with this partner, and I do plan to work with her again. She has strong background knowledge and is hardworking. She was able to solve one problem we had on our own, and was able to show me quickly how it worked. She was able to help figure out how to properly use some functions and even some which puzzled me she was able to explain them. Neeka also was extremely eager and excited to get started on this project which I believe is important to have in a partner. She was well versed in programming as well, and knew some programming tricks which were quite useful when we programmed together.

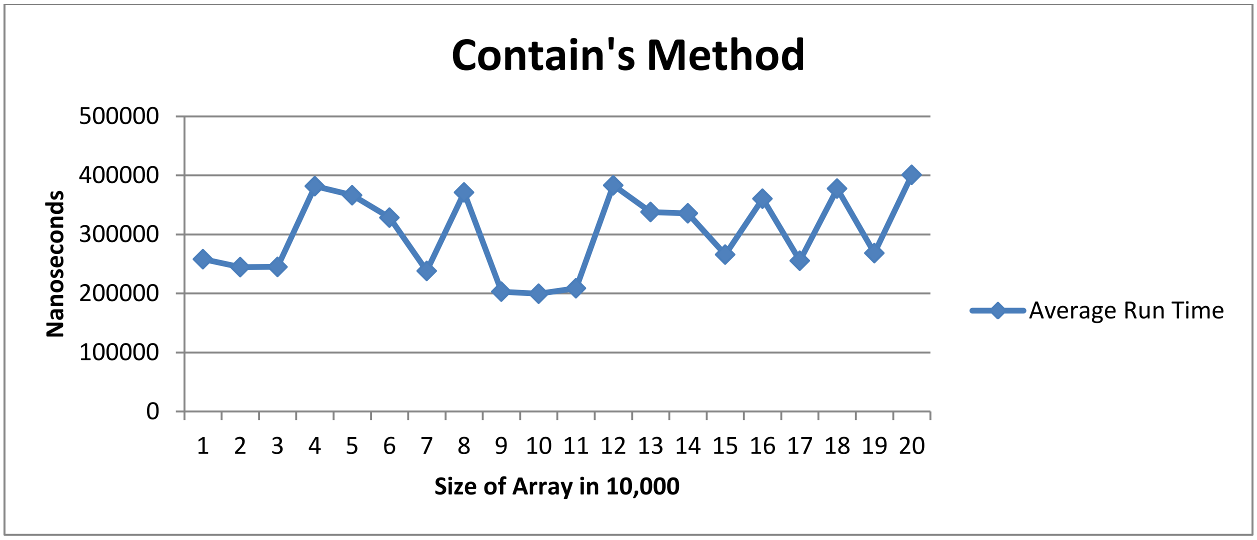
4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development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

If we had used the Java List Arrays instead of normal arrays, I believe that our program would have run faster. The reason for this is that the time that used to call the grow array begins to use up more and more time with the increasing number of objects being inserted. It also sorts the list for us. This would have saved more time when the sorting function was called. This would have saved a considerable amount of time since than every time we had to insert a new object we would not to have to worry about moving around the array to ensure that it was being placed in the correct position. Another positive part of this would have been that the program that Java ArrayList utilizes is extremely efficient, and would have saved us the time writing the sorting, and growing functions. This would have made the development of the program considerably quicker. It would also have run faster due to the greater efficiency of Java ArrayLists. Overall I believe that using a Java list would have been better

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

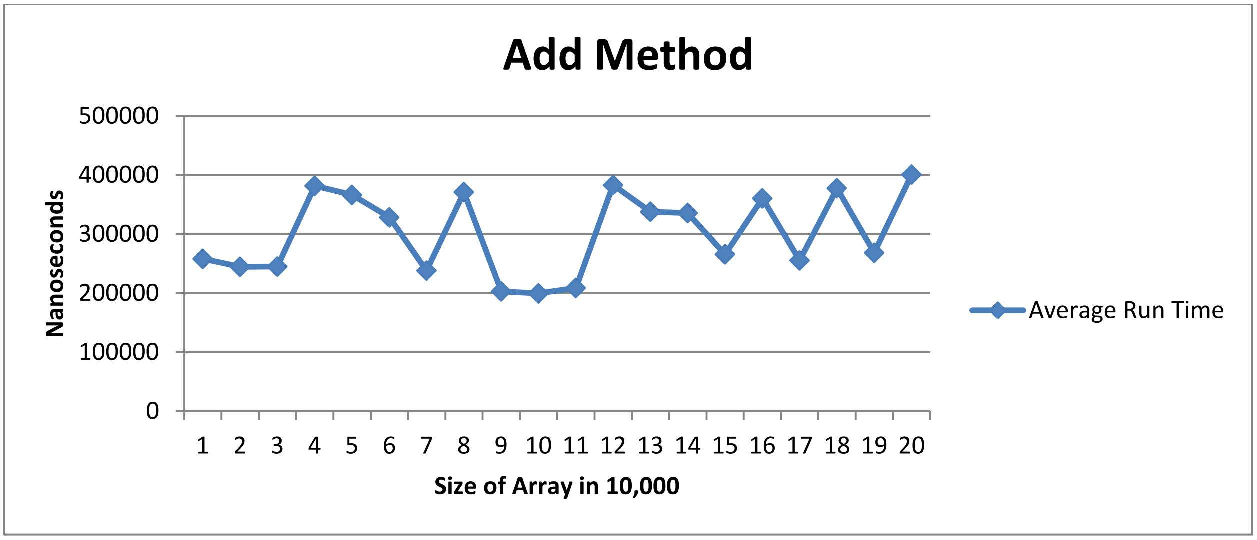
The Big- O behavior should be a linear line of value N. Since the contains method has one for loop it goes through all of the variables within the list at least once if it is a worst case scenario. As more items are entered the longer the average run time will be for the objects. Thus the growth within the graph should be steady and in a somewhat linear line as more the Array is increasing with size.

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?



It follows my prediction in certain parts as it is seen to grow when comparing the lows within the graph, but increased as the arrays got bigger. Overall though my graph was not as nice as I had predicted. There were some extremely low times that were lower than when the array was smaller, and similarly with the higher areas. This led to my first estimation to not necessarily be a straight growing growth line. Nevertheless overall the larger the growth the longer it took the longer amount of time it took to check. This graph would have displayed an N type of graph had the distances between the array sizes been larger.

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)?



Normally I believe that the efficiency of our program would be N^3, but sometimes such as when the Array needed to be increased it would be N^4. N^3 would be used if we are only adding an object and sorting it. The worst case scenario would be that the program would take N^4. Which would mean that the program needed to grow, and the number that is being inserted is at the beginning of the list since all of the subsequent objects in the array would be needed to move one down, and that it had no duplicate so the contain method would have to look through all of the objects to ensure it is not in there.

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

My partner and I had spent approximately 9 hours on this assignment. This included testing and debugging as well in order to ensure that what we had was running correctly before we proceeded.