Jacob Osterloh
U0645881
Analysis Document
Assignment 3

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

   Jordan Davis. Jordan Davis will be submitting 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?

   We switched about every hour. I felt like it was a good routine because it allowed us to get into a good groove, but it wasn't too long that we got distracted or lost.

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

   Jordan was great, we had to spend a lot of time on this assignment and we were both able to schedule our time very easily. Yes, we plan on working together 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.)

   If we used a list we wouldn't have to worry about resizing the set because it would automatically resize if new elements were added or removed. This would have made our code much simpler during program development time, because a lot of the issues we ran into concerned the size of our array when adding or removing elements. But because the basic array we created dynamically expands, not as much time is taken to expand the array as compared to a list, which must expand or contract depending on every element added or removed, but overall the running time efficiency between a basic array and a list is minimal.

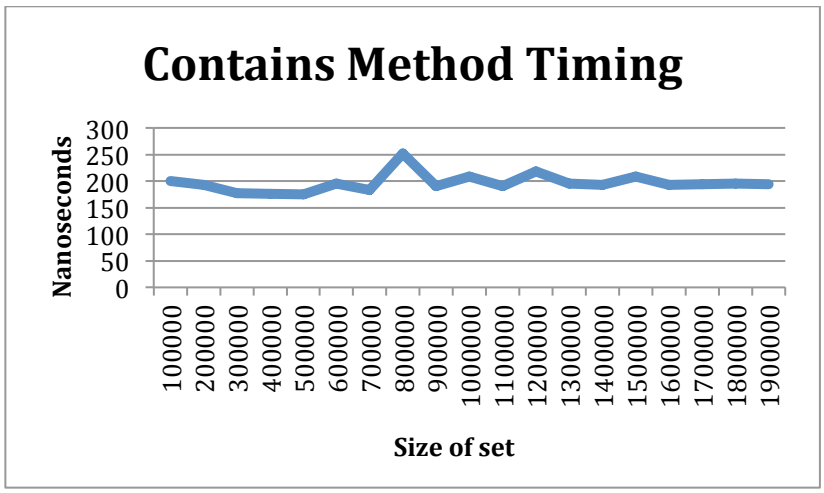
5. What do you expect the Big-O behavior of MySortedSet's contains method to be and why?

   I would expect the Big- O behavior to be N because we are using a binary search in our contains method, so the time and the amount of steps to search

through a set should not increase dramatically even when the set size is very large.

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?

   Somewhat matches what we predicted, but looks to be more like the Big-O behavior logN than just N, so it looks more efficient that what we previously assumed.



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)?

   Just finding the location to insert an element is only logN because of the binary search method used to find where an element ought to be inserted. Our add method handles the reordering and resizing of the set which worst case, takes much longer because it is of order N^2.

8. How many hours did you spend on this assignment?
Probably about 19 hours… hard hours of testing, debugging, and chillin' with the TA's.