Cole Wilkes
Assignment #3
Analysis Document

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

Patrick Egan Anderson. Patrick Egan Anderson.

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 every couple methods. It was a good amount of time for navigator and driver, I don't think I'd prefer more or less.

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

He's way smart and a hard worker. I plan to 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.)

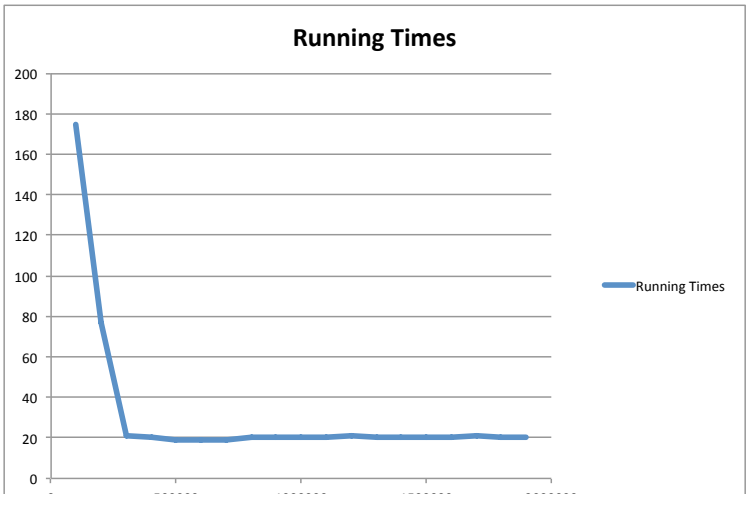
Using a java list would have cut development time down by quite a bit because we wouldn't have had to implement a lot of the methods. It would have been less efficient for run time because the methods a java list has do not use binary search, and would require looping through the entire list.

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

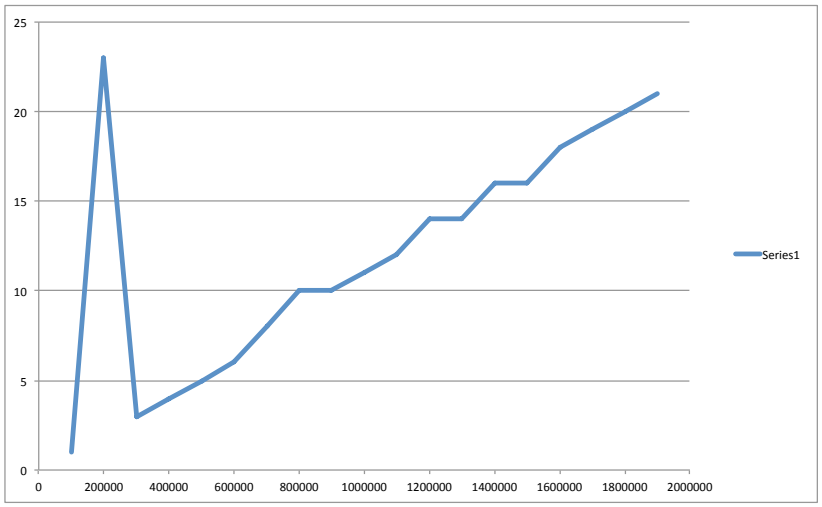
Log N since there is a binary search done on the object (N) that is passed to contains.

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?

No the growth rate does not match our predicted behavior.



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)?



Add method bigO notation : O(N)

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

20 hours spent