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?

Brennan Myers. I turned in 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 all the time. Just whenever someone had an idea they took the keyboard. It worked really well.

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

Brennan is a great partner. I plan on working with him for this next assignment too.

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.)

Java lists are already sorted so we wouldn't have to use the binary search that we created. This would make runtime much more efficient.

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

(LogN) because it is only calling the binary search and returning true or false.

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 seems to match the big-Oh behavior from what I can tell. We plotted it on our graph and though our tests were small the biger numbers still had fast run times.

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)?

The worst case scenario for binary is O(logN) and that would be if the element had to be put in the beginning every time so it would still be O(logN)

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

About 16 hours.