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?

   Katherine Liu is my partner. I will be submitting the assignment.

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

It was split pretty even. I'd say 50/50. I am completely fine with how I work with my partner. We switch relatively often. We could switch less often for more time in a specific role but, it always balanced out in the end.

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

Yes, I enjoy working with my partner we are on equal skill levels so neither one of us is carrying the other one.

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.)

The add/remove methods would have been a lot simpler. I believe it would have been more efficient running time and development time. Lists can change size a lot easier than arrays can. I am also more familiar with lists than arrays (although that is not excusable.)

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

I expect it to be O(log(N)) because it is a binary search. This method will cut the set in half each time it looks at it. By definition that is log base 2. It would take twice as many items to make the algorithm do one more unit of work.

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?

   Yes is does match my prediction.

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)?

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

It took about 15 hours but maybe more (worked on it 4 separate days.)

Programming partners are encouraged to collaborate on the answers to these questions. However, each partner must write and submit his/her own solutions.

Upload your document (.pdf only!) to the Assignment 3 page by 11:59pm on February 5.