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?

Anna Buchman. I submitted 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?

I would say anywhere between every few minutes to every 15-20 minutes. Again, we switched whenever someone got an idea and didn't switch until we got stumped and the other person came up with a new plan, which gave us about equal time to work. I am happy with the amount we switched.

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

Anna is very good at picking up the little stupid mistakes I made, and is very proactive – while I'm typing she's looking up things that will help us, researching as I am, and a lot of times she's broken us out of a stump. I plan on working with her as long as I can.

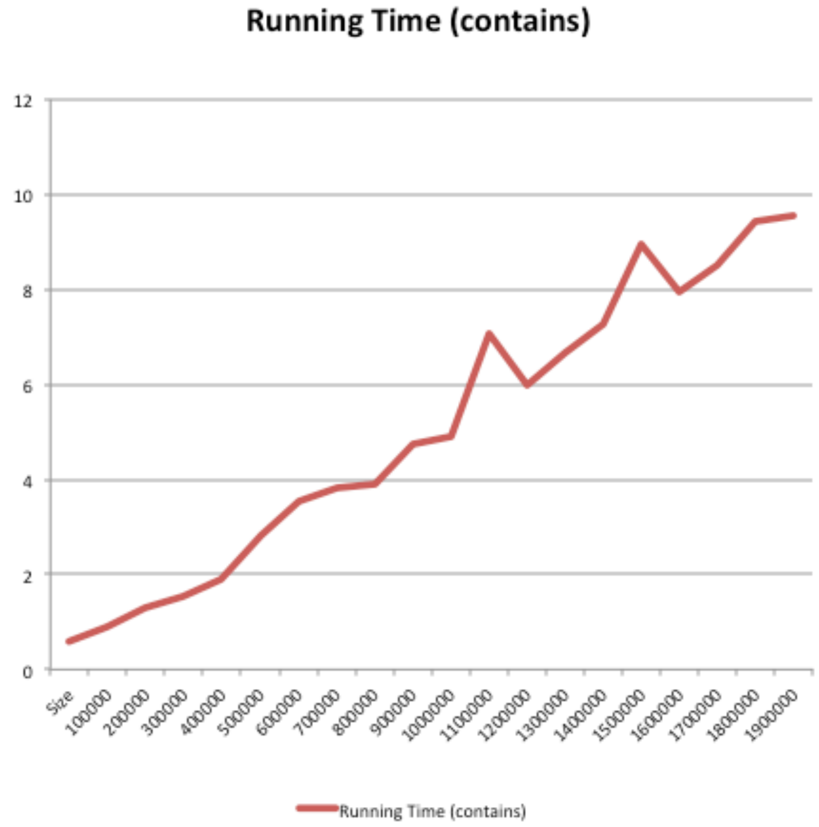
4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development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

We wouldn't have had to worry about redoing the array, moving items, etc. with a Java List, and it would have an already implemented the containsAll method that already went with it. It would have been more efficient.

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

It uses a binary search, so the Big-O behavior/complexity is N log(N).

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?



Given more times (and patience) it looks like it would fit the logarithmic model, yes.

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?

~10 hours