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

Filipe Wesley is my partner and 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?

Not so often. We preferred to switch role less often because wanted to finish the program more quickly and he has more experience with Java programming than me. I have more experience with C and C++.

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

He was perfect as a programming partner, besides working he even answered a lot of questions I have about Java.

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 List is a little easier to code than arrays, more flexible as well. In our program some points would change drastically as the point we needed to change the length of the array, because the List don't need we do that manually, it has an iterator, etc. With Java List we would have more effiency in both running and development time.

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

The Big-O would be the case that the set doesn't contains the object, because it would need to do several binary searches until the end to see that the object isn't in the set.

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, it matches.

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?

About 7 hours.