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

My programming partner is Nobuhito Nishihara. Nobuhito Nishihara submitted the source code of my program.

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

Me and my programming partner switched roles on an hourly basis. I would've preferred to switch more often because I tend to feel lost with where I'm going with my code after working on the code for more than thirty minutes in a row.

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

Just like with the first assignment in the class that I had with him, my programming partner is an extremely responsible person who gets as much TA assistance as possible when he struggles with the assignment rather than copy off of solutions found on the Internet or from other programming groups. His knowledge of the Java language is relatively equal to mine so 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.)

If I had backed the sorted set with a Java List instead of a basic array, the main points in which my implementation would've differed would've been that less JUnit tests would've be en used, I wouldn't have to expand the size of the list on a regular basis as more entries are added, and the list would be able to work with any data type without having to use the comparable and comparator interfaces. I expect that using a Java List would be efficient because the basic array functions in the same way as the majority of memory-consumptive program languages while the Java List uses special lines of code to reduce the worst-case Big-Oh complexity degree, develop the program quickly, and run the program quickly.

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

I expect the Big-O behavior of MySortedSet's contain method to be log n because it uses binary searches in a single for loop.

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?

The growth rate of these running times doesn't match the Big-oh behavior that I predicted in question five.

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)?

For an element not already contained in the set, it takes a constant amount of times to locate the correct position at which to insert the element. In the worst case, it takes log of n times to locate the position to add an element.

8. How many hours did you spend on this assignment? I spent over forty hours on this assignment.