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

My programming partner was Vinicius Goncalves de Carvalho - u0973889.
He submitted it.

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 very often. I'm okay with it, because I enjoy coding.

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

10/10. Great partner. 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.)

Running time efficiency wouldn't change much, but program development time would change drastically. It would be a lot easier since Java Lists have dynamic size changes and associated iterators, and other minor stuff that would help in the moment of coding.

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

Log(N). Because it's basically a recursive binary search, which cuts the size of the array in halfs.

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?

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?

Around 7 hours