

Ellen Brigance  
U0795545  
CS 2420  
February 5, 2015  
Assignment 3 Analysis

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

Aidan Tarufelli was my programming partner. He also submitted our 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?

We didn't switch very much. I wish we would have switched more because he drove most of the time so I felt like I had a very passive role.

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

Other than the fact that we rarely switched roles during this assignment, I do enjoy having my programming partner because he is always willing to work on our assignments with me for as long as it takes to get them finished.

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.)

Implementation with a Java List would have been more efficient because its class contains almost all of the functions we had to write on a very basic level. In regards to having an actual sorted list, we would just have to override the Java List functions to operate under our preferred conditions.

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

The Big-O behavior should be  $O(\log(N))$  because contains uses a function that does a binary search, which cuts the number of elements that we have to search through in the array in half every time it loops.

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?

Altogether, we probably spent around 20 hours on the assignment.