Brantly Walker - CS 2420 spring 2015

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

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

During this assignment I was the driver, I think it is good practice.

3. Evaluate your programming partner. Do you plan to work with this person again? Yes I will be working with my partner again, although my partner doesn't understand how long the programs take to write and we didn't get started until Wednesday because she had other things to do during our normal programming time and was sick. And the partner programming rules dictate we only work together..

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.)

I think it would have been more efficient, the main reason being that when we run out of space we don't need to copy over items to a new array.

5. What do you expect the Big-O behavior of MySortedSet's contains method to be and why? I expect it to be logn for the contains method, but nlogn for the containsAll method due to the double nested 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?

I couldn't figure out how to do that graph :/ The growth rate I figure would look typical of big O logn. Even when the size gets larger because it is only increasing by a small amount because the binary search method halves it every time.

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)?

It should take n/2 times to find the correct location using the binary search method. Big O of the worst-case is Logn.

8. How many hours did you spend on this assignment?

We spent about 6 hours Wednesday, and 6 hours Thursday.