Cody Woolsey
Assignment 3 Analysis

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

My programming partner was Gage Glenn. Gage 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?

We switched quite often, I'd say about every 10 or 15 minutes. I feel the amount we switched off was appropriate for the assignment, and that we were both able to do about equal work. Not only that, but switching as often as we did kept us both engaged and helped us stay on track.

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

Gage is a really good partner for me, since we both have similar skills and work at similar paces. He's really devoted to the class, and we absolutely put equal effort into our assignments. I've worked with Gage these first two assignments, and plan to continue working with him until we are required to switch partners.

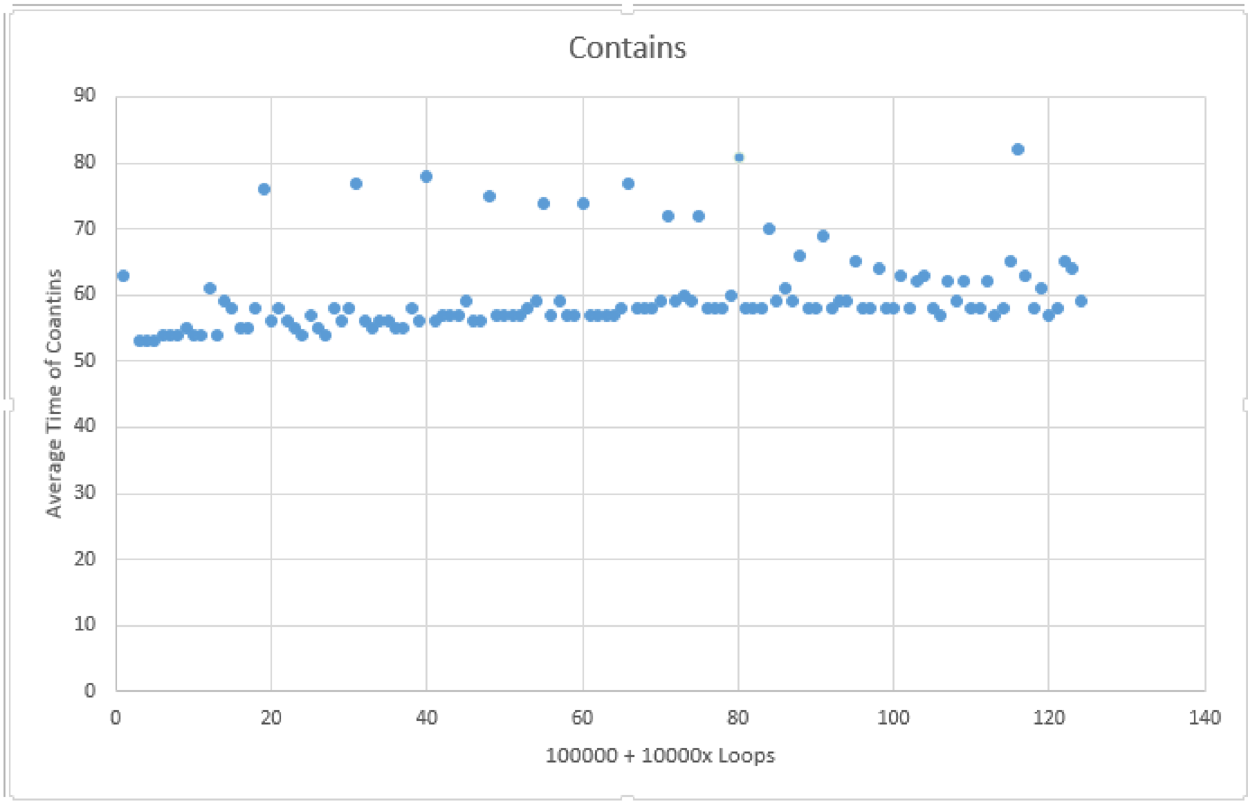
4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development time.)

The implementations would be largely the same, besides using some of List's default methods to your advantage. For example, the shifting of elements in the add and remove methods are taken care of in the List's add and remove methods. As far as efficiency goes, a regular list is probably backed by a basic array, meaning Java is performing similar operations on the list as we would. Because of this, an implementation with a List would probably not be much more efficient.

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

The Big-O behavior of the contains method should be O(log(N)), since the binary search algorithm follows the rule of halves, and is essentially a single boolean calculation for each iteration.

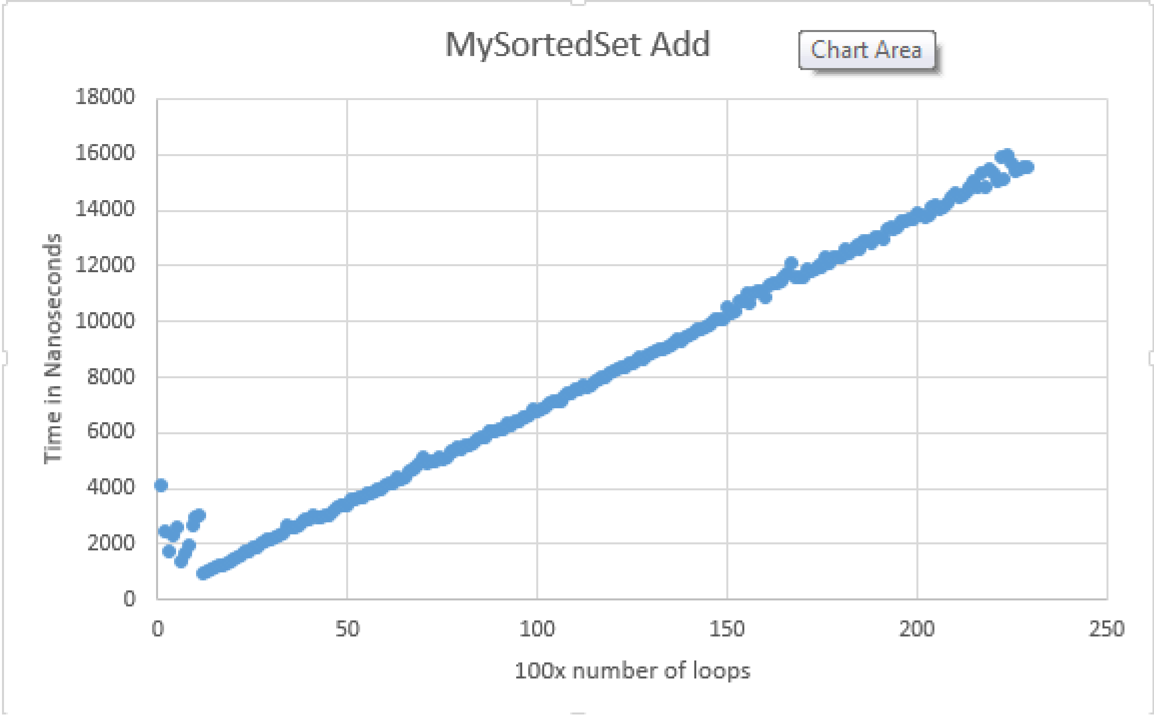
6. Plot the running time of MySortedSet's contains method for sets of sizes 100000 to 2000000 by steps of 100000. Use the timing techniques demonstrated in Lab 1. Be sure to choose a large enough value of timesToLoop to get a reasonable average of running times. Include your plot in your analysis document. Does the growth rate of these running times match the Big-oh behavior you predicted in question 5?



The growth rate is consistent with a O(log(N)) algorithm.

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)?



The complexity for the add method to determine where to insert the value is O(N).

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

We spent about a total of 7 hours on the assignment.