Samuel Bridge
U0984334
CS2420

Assignment03 Analysis

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

My programming partner is Osama Kergaye, he submitted the source code of the 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?

We didn't switch roles often. Most of the time it was Osama writing the actual code on his computer while I oversaw and collaborated. The only times we would switch is when there was something like a bug that he didn't understand and I would take over to help explain and write the code myself. I do not prefer to do this any differently as we were very efficient in our work this way.

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

My programming partner is proficient in Java, at about the same level as I am and was very organized and time efficient. I do plan to work with my partner 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.)

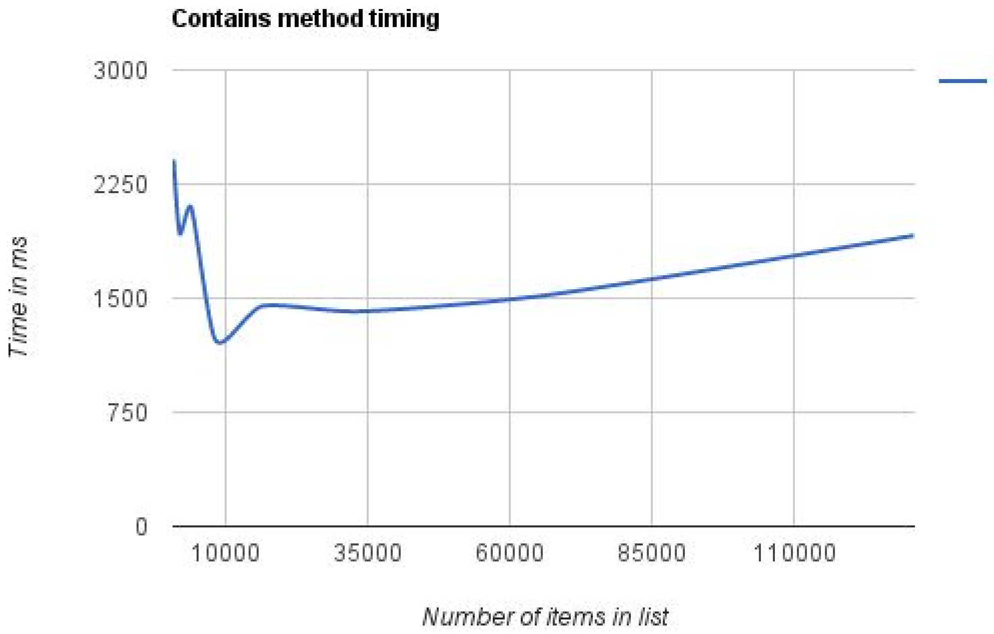
If I were to have used a java list instead of a basic array the implementation would have differed because java has built in functions for binary search and the other methods I implemented. It would have been much more efficient in terms of development time but in terms of run time I don't see any ways in which the running time would have differed. This is because Java's built in list uses binary search just like ours does.

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

I expect the Big-O behavior to be log n. This is because binary search works in a logarithmic behavior, taking half of the list each time.

6. Plot the running time of BinarySearchSet's contains method, using the timing techniques demonstrated in Lab 2. Be sure to use a decent iteration count 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?

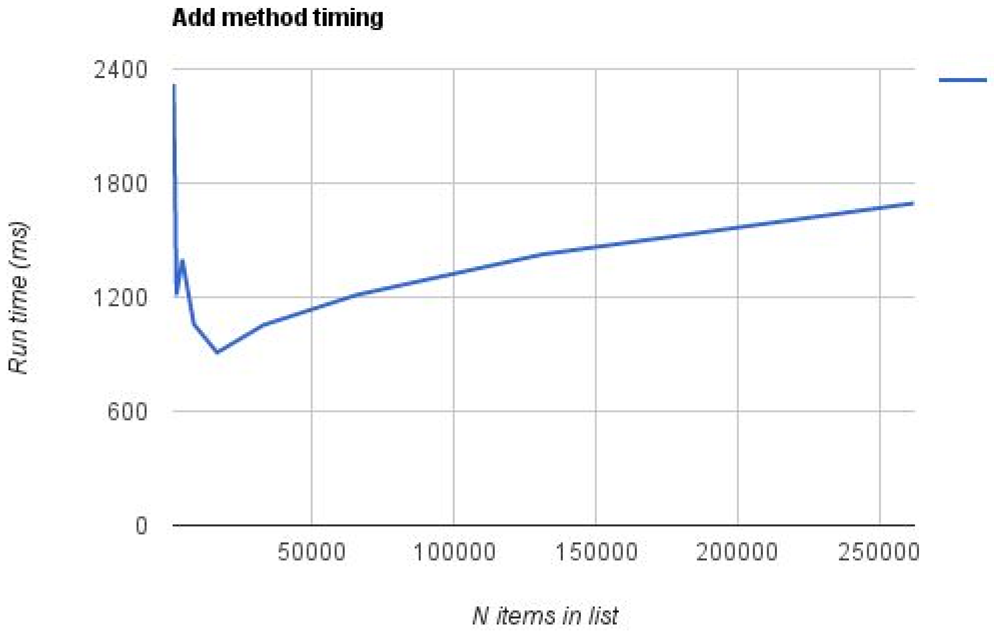
In the first couple times of testing the contains method the time is negligible. After about the third list size it is apparent that the Big-O behavior is in fact logarithmic, as stated in question 5.



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? 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 behavior of the add method is also log n, likewise with the worst case scenario. This is modeled in the graph below.



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

I would say a total of around 15-20 hours were spent on this assignment.