Clayton Hislop
uID: u0515744

Assignment 3 Analysis

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

My partner was Richard Child and he submitted our code for this assignment.

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

I would say we switched about every hour or so. I think we switched a good amount but sometimes we would get stuck with a bug or some other issue and the person with control of the keyboard may not have been able to do a lot of coding if we didn't solve this issue before switching again.

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

I think I lucked out with getting Richard as my partner. He seems to have more experience with programming than I do so that definitely made it easier for me during the assignment. It was also nice that he stayed until we finished our work even though he was risking missing the train home. I also appreciated him doing the work with Excel since I do not have a lot of experience using that application. I would be very happy to work with Richard again.

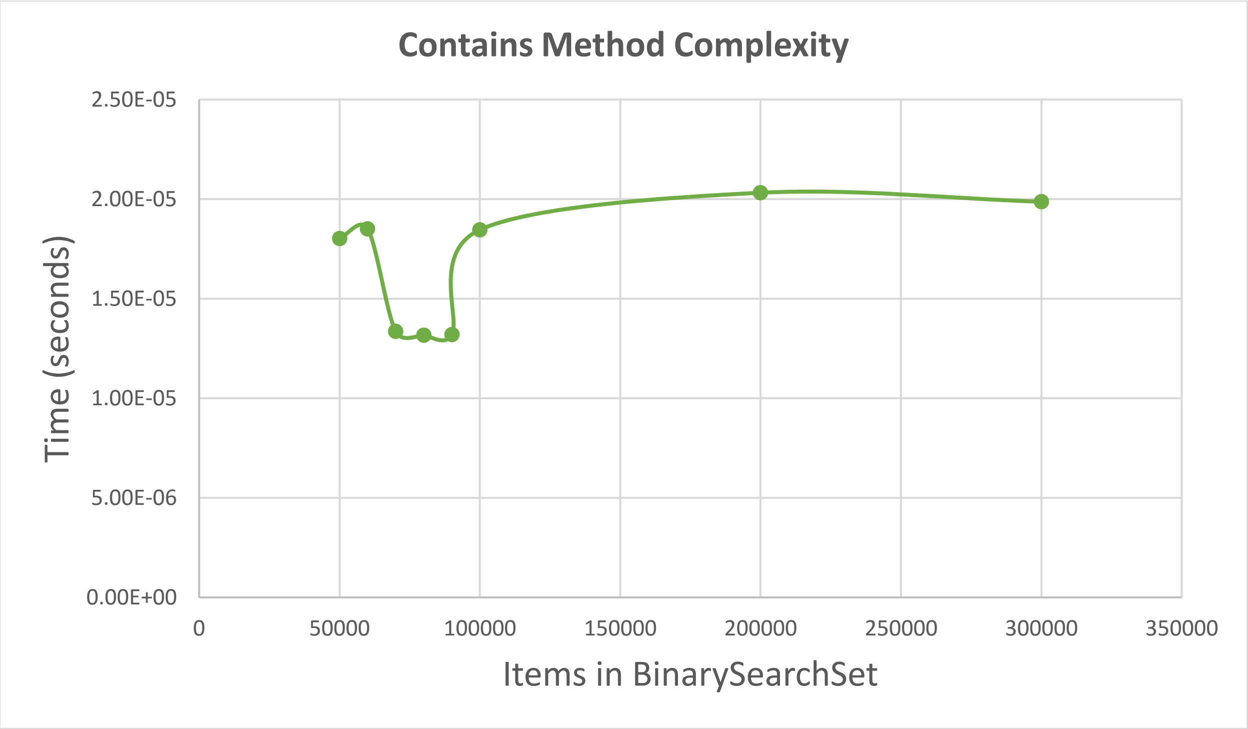
4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development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

If we had used a Java List we would not have needed to write any of the methods we were asked to write as they all come pre-built in Java Lists. And because sets cannot have duplicates and lists can, we would not have had to check if an item was already part of the list. Because the methods are already implemented in the Java List, it would definitely be more efficient to use a List instead of writing our own methods. As far as running time is concerned, it is probably faster to use the Java List as well because I'm sure I don't write more efficient code than the developers of Java.

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

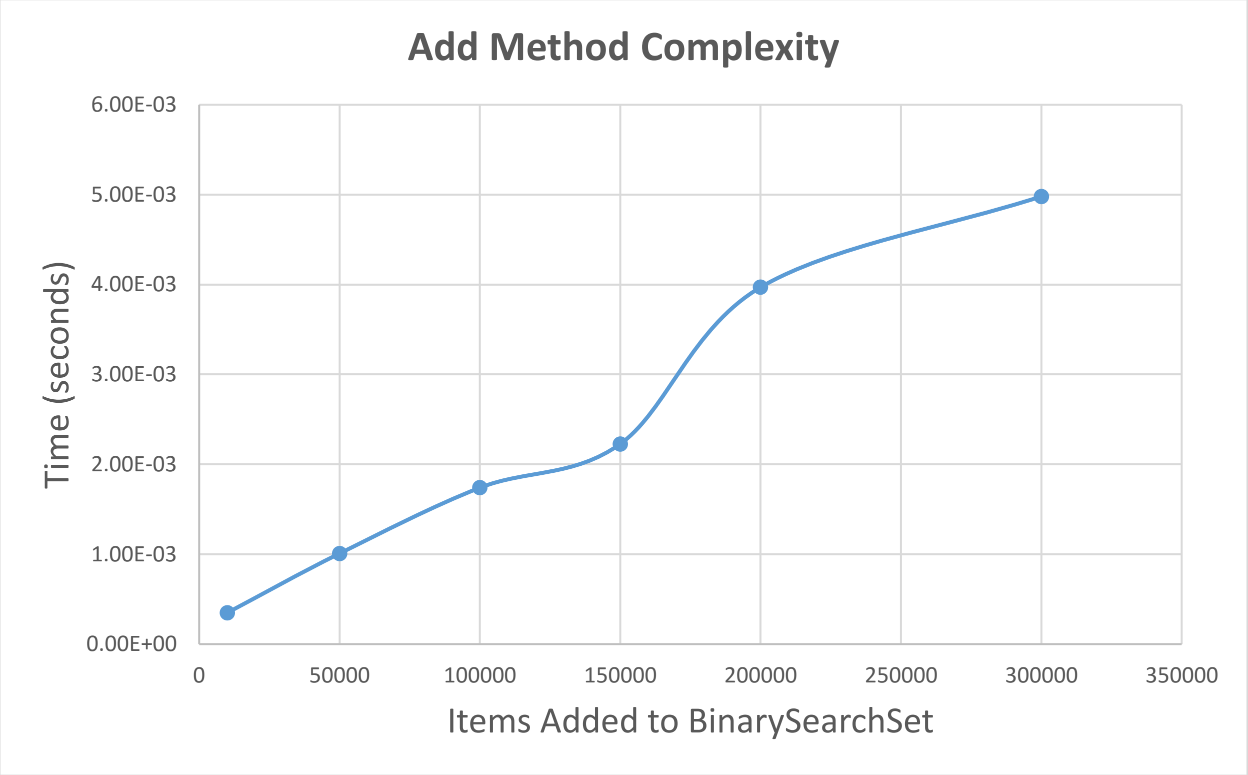
I would expect the contains method has a O(log N) behavior. Because we are cutting the searchable items in half each time we make a comparison and don't find our target, we are following the repeating halving principle.

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?



For this timing test we used various sizes of Sets and iterated over each Set 1000 times using a new random number each time. I don't know if it is because we used a random number each time, but the graph does not look like what I was expecting it to look like.

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., iteration count), 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)?



As mentioned in question 5, when we don't find the item we are looking for we cut the number of items to be searched in half, making it a worst case of O(log N).

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

I would say I spend about 11-12 hours on this assignment, about 9 of which were spent working with my partner.