Richard Child
u0581030
CS2420

Assignment 03 – Analysis

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

My programming partner is Clayton Hislop. I submitted the source 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?

We switched roles as driver and navigator roughly every thirty minutes. I think that is a good amount of time in one role for our partnership. By switching roles that often I felt like different parts of my mind were being used to solve the problem. The frequency was enough to keep us engaged but not too much to be disruptive.

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

Clayton is a very good partner. He listens and provides feedback. He isn't afraid to ask questions and critique our own work. He was always on time when we met and he carried just as much of the work load as I did to complete this assignment. We will definitely be working together again this semester.

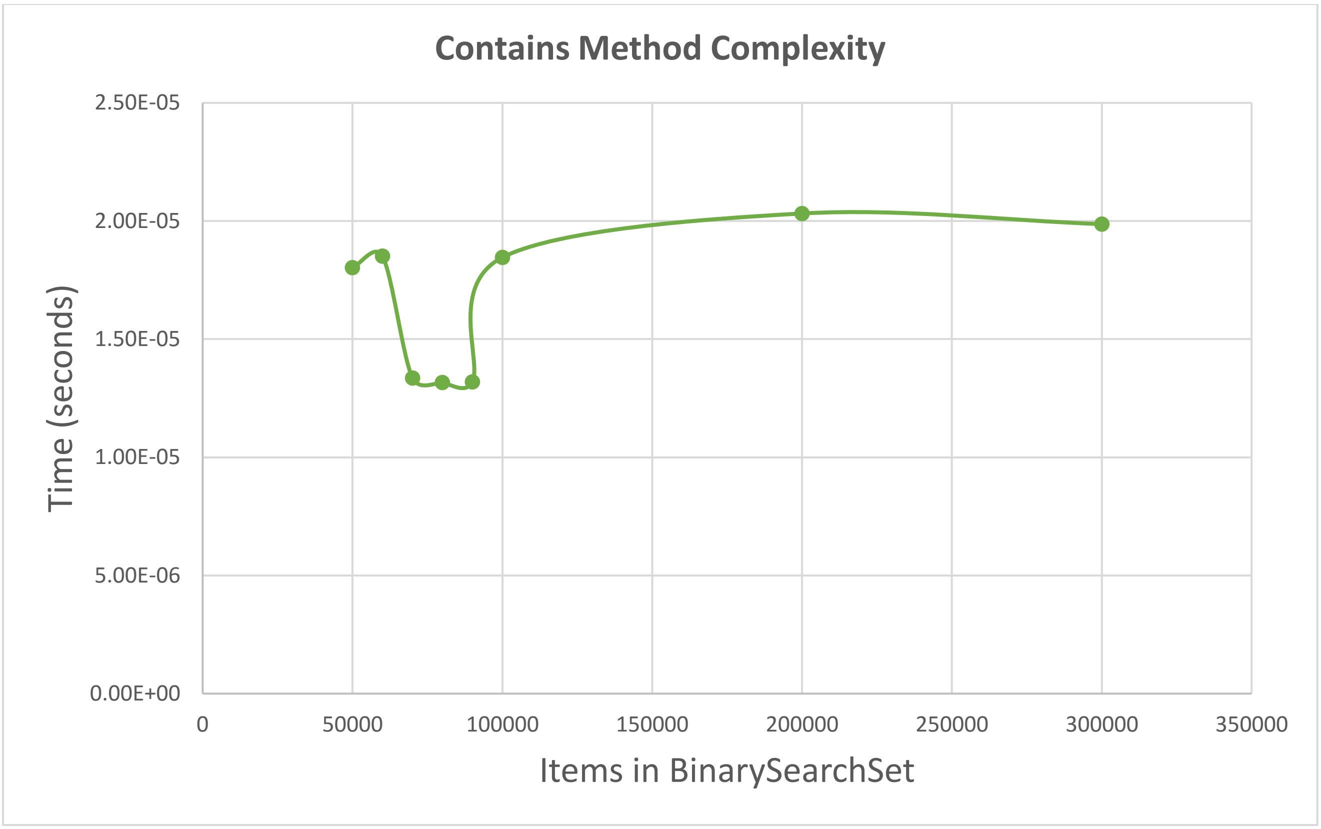
4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development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

If we had been able to back our BinarySearchSet with a Java List the developing process would have gone by much faster. We would not have had to implement our own remove and add methods, which were the most difficult parts of the assignment. It is true we would have had to develop the binary searching methods but all the data shifting would have been easier. I believe the run time efficiency would have been similar since Java's own collections are backed by basic arrays which use linear complexity copying methods. The main difference would have been in actual developing time.

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

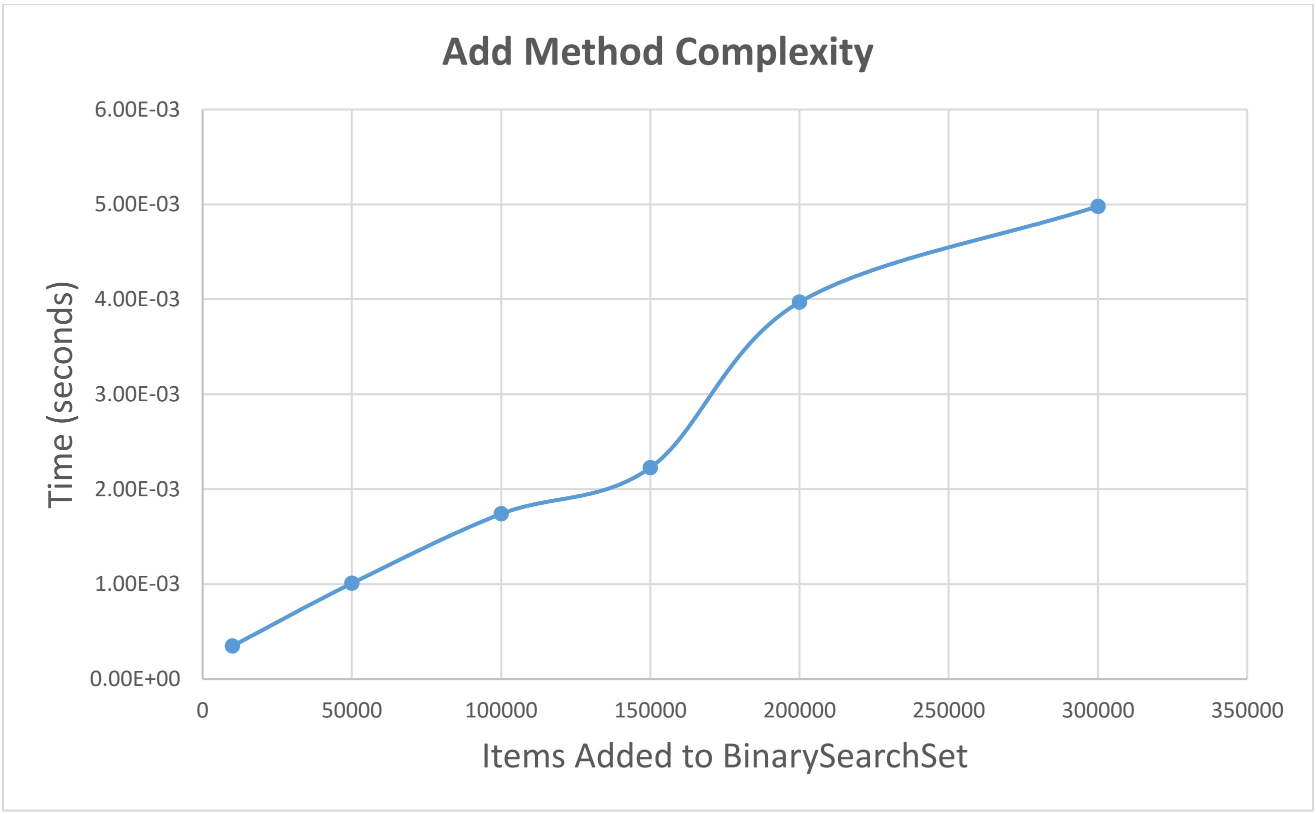
I believe the complexity of the contains method should be logarithmic. This is because it uses a binary search algorithm (one with repeating halving) to locate the item. After locating the item the contains method just returns it.

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?



We timed the contains method using different sizes of BinarySearchSets ranging from 50,000 items to 300,000 items. Every search element (the element to look up) was generated at random before starting the timer. The data we collected was somewhat perplexing because the time only increases very slightly from 100,000 items to 300,000 items. At first I believed the method was constant but by looking at the graph we can see a definite logarithmic curve.

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)?



To locate the correct position in which to add a new item, I believe would be a logarithmic time relation to the size of the set. This is because we are just using a version of the contains method to find this index. When we set up and ran timing tests for the add method we found that for the most part the complexity is linear. This is surprising because just by looking at the code the worst case is O(NlogN). Perhaps the worst case does not occur very often.

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

My partner and I spent approximately 20 hours on this assignment.