

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

My partner for this assignment was Alex Ljubicic, and he submitted the code for our 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?**

About every hour or so, or if one of us had a better understanding of what we were currently doing. The amount we switched worked out well for us,

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

Me and Alex worked well together, despite never meeting him before we got along well, were able to meet frequently and he was quite flexible. He was helpful on the assignment and his creativity helped us greatly to solve the problems. I will work with him again on the next assignment and likely the final assignment.

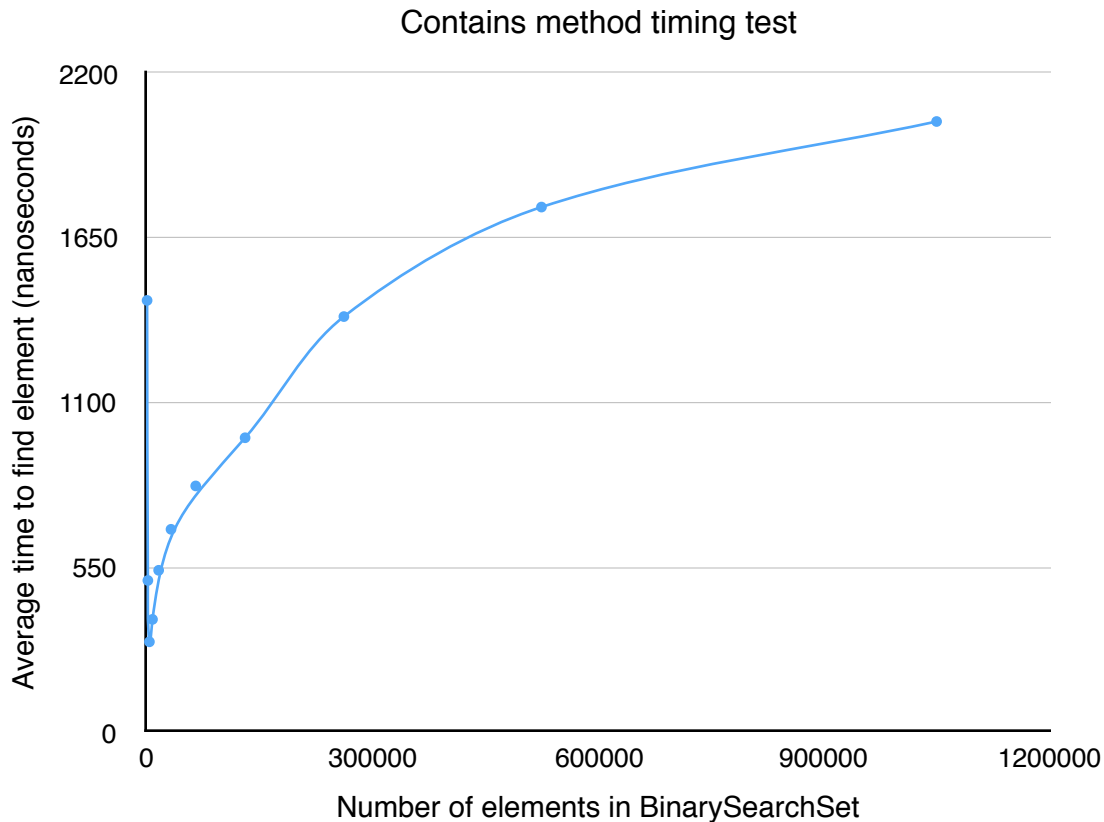
**4 If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development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

Due to the flexibility of Java List and the already available methods within it, it would have been much more efficient in term of program development time. Many of the methods would have to be altered only minorly and require half of the length of the current assignment. In terms of running time efficiency, it could possible be more efficient, because it would not require the extra time it takes to double the size of the basic array (although the list would do that within itself, it would surely be more efficient). But I expect any differences in terms of running time would not be massive.

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

I expect it to be  $O(\log n)$ , because we use a binary search for the contains method which only has to increment by one more search when the array size doubles.

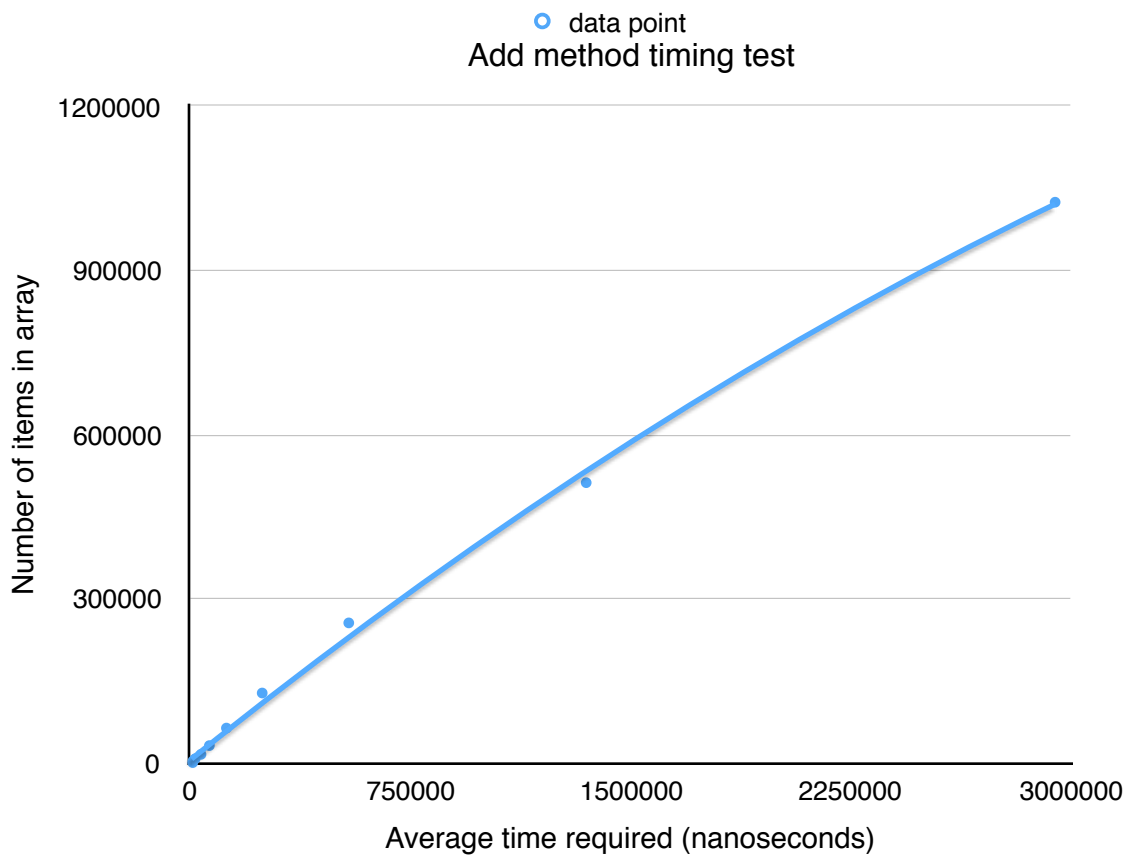
**6. Plot the running time of BinarySearchSet's contains method, using the timing techniques demonstrated in Lab 2. Be sure to use a decent iteration count to get a reasonable average of running times. Include your plot in your analysis document. Does the growth rate of these running times match the Big-oh behavior you predicted in question 5?**



Yes, the behavior of our graph closely matches the  $\log N$  big O behavior which we were expecting

**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. 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 takes  $\log(n)$  time to locate the correct position every time. This will be the same for the worst case, as it will always be the worst case the way we wrote it. Different from contains, it always takes the maximum time because you must narrow it down to one position by testing the positions directly above and below it. Unfortunately the add method is slightly more complicated than the contains, so it takes slightly longer and the actual time is slightly bigger than  $\log(n)$ , but the closer you approach infinity the closer to  $\log(n)$  it will be.



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

This was a pretty long one, I spent around 20 hours in total.