Adrian Bollerslev

Then you are satisfied that your program is correct, write a brief analysis document. The analysis document is 30% of your Assignment 3 grade. Ensure that your analysis document addresses the following.

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

Eduardo Ortiz. I 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?

I think we distributed the work fairly well. Although I initially started with more it eventually evened out. I did most of the coding on the class while he did most of the coding for the tester.

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

I would work with him again we were good at splitting up the work and working as a team.

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.)

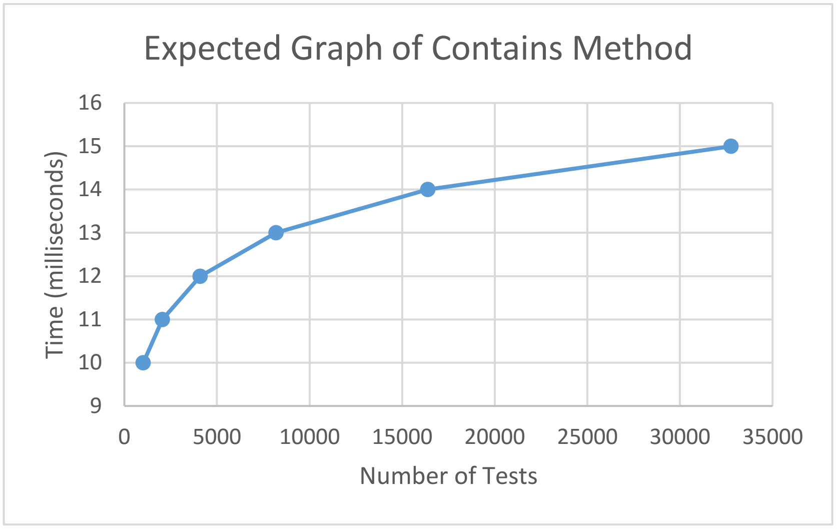
I believe a Java List would have been much more efficient and easier to code. The list comes with methods for sorting and another increased functionality compared to an array. In general arrays are good in cases where the data is a fixed length but in this assignment we were adding and removing data constantly.

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)). This is because a binary search works through midpoints. If we quadruple the amount of data we will only have two more steps and so on. One can think about a binary search as looking up a word in the dictionary. We do not have to look at every element, simply look at the middle and whether it is before or after the word we are currently looking at.

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 could not run the code we had so this is expected data)



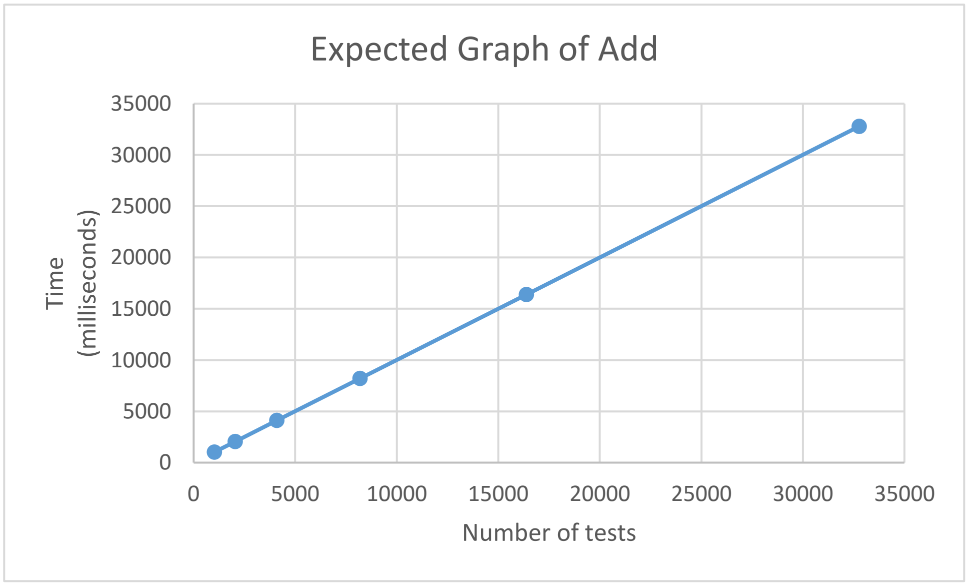
Number of Tests	Time(milliseconds)
1024	10
2048	11
4096	12
8192	13
16384	14
32768	15

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)?

(We could not run the code we had so this is expected data)

I estimated O(logN)+N. This is because the contains method which is part of the array takes O(log(N) as described above. Then we must traverse the area one more time to correctly insert the Object.

Number Of Tests	Time (milliseconds)
1024	1034
2048	2059
4096	4108
8192	8205
16384	16398
32768	32783



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

15 hours.

Programming partners are encouraged to collaborate on the answers to these questions. However, each partner must write and submit his/her own solutions.

Upload your document (.pdf only!) to the Assignment 3 page by 11:59pm on February 5.