When 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?

My programming partner was Martin Izdimirski. My partner 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?

We switched about every 30 minutes. I would have preferred to switch a little bit more often because the coding time between us was a little imbalanced.

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

We worked well together and helped each other come up with ideas. I plan to work with him again.

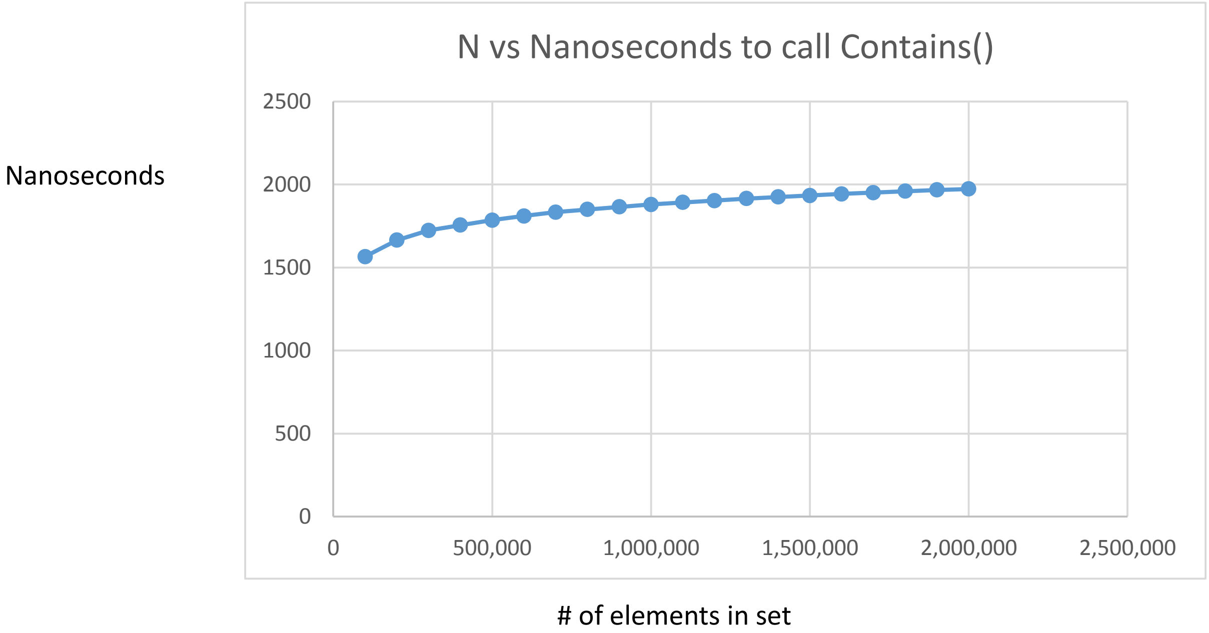
4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development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

If we used a Java List like an Arraylist, then our implementation would have differed slightly. In our add function we wouldn’t have to have code to grow the array. Also we would already have an iterator. I expect that a Java List would be more efficient. I think that running time efficiency would be the same as an Arraylist is an Array behind the scenes. Program development time would have been more efficient as we wouldn’t have to code as much.

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

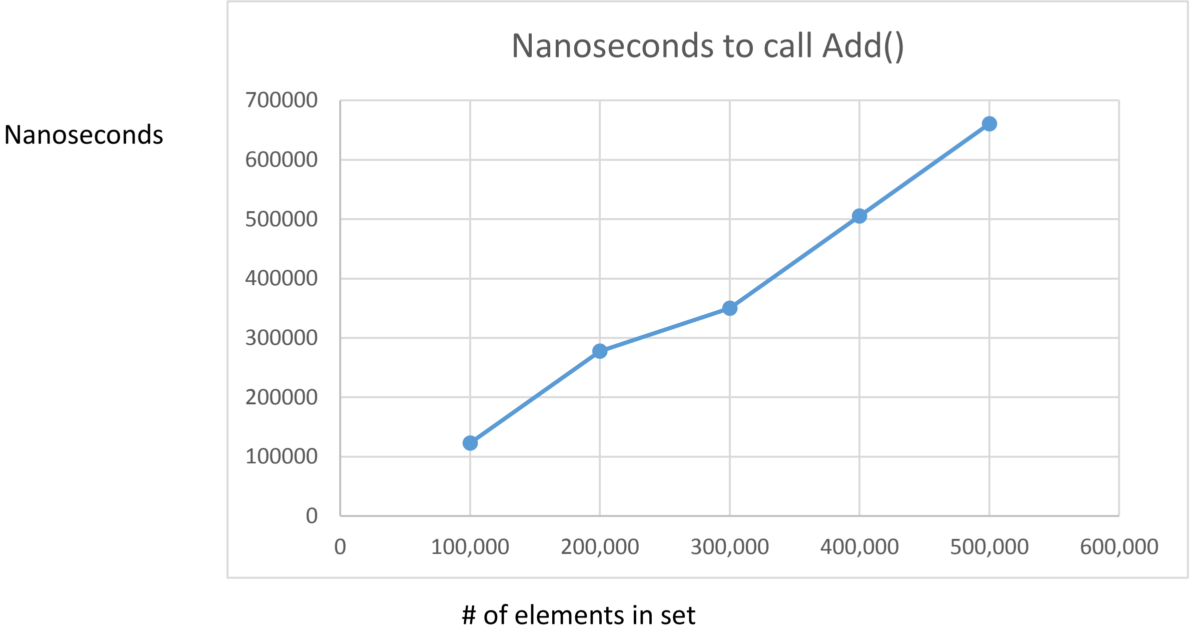
I expect it to be O(log(N) as it is essentially a binary search that is looking for one element.

6. Plot the running time of MySortedSet's contains method for sets of sizes 100000 to 2000000 by steps of 100000. Use the timing techniques demonstrated in Lab 1. Be sure to choose a large enough value of timesToLoop 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 growth matches my answer in question 5 as the plot looks like logarithmic growth.

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., timesToLoop), 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)?



This graph is linear because our add function’s complexity is N + log(N). The N is from remaking the array and log(N) is from searching where to put the new element. So add is O(N).

We use a binary search to find where to add the element. So the worst case for that is O(log(N)). So the worst case to find where to locate the position to add an element is O(log(N)).

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

About 12 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.