Morgan Mercado
U0627613
Assignment 2 analysis document

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 Brandon Mercado, I will be turning in 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 every 30 minutes. I preferred navigating because I like to think about where we need to go with out code.

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

Brandon was an exceptional partner again. He was well prepared to handle the instructions we were given in the assignment page and corrected many coding errors I had. I will work with this person 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.)

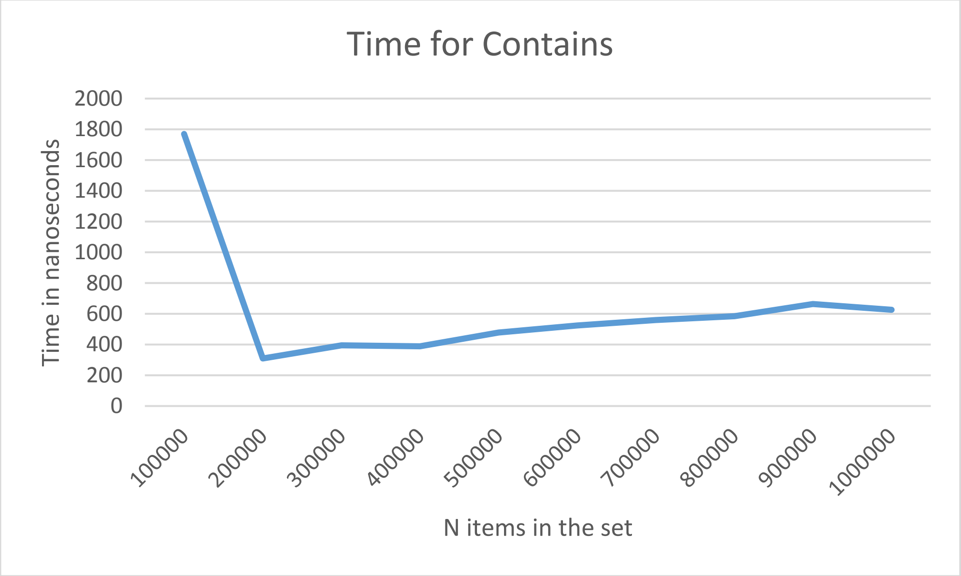
I expect that using a Java List would have the same efficiency as using an array. Only a few syntax would have to be changed to get MySortedSet to work with Java Lists.

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

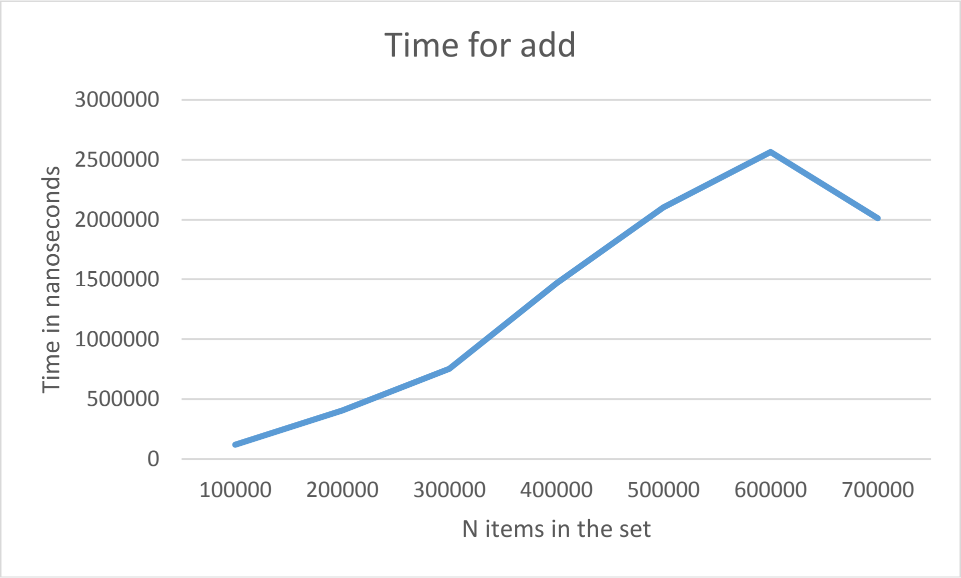
I expect contains to run at O(Log(N)) since we used a binary search.

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?

The beginning N timing is unusually large but I expect that to be because java had to do some warming up initially. It then shoots down to a very low timing for contains and grows at a very slow rate looking to be at the expect timing of Log(N).



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)?



It takes quite a bit longer to find the correct position for an item then it does to find a specific item.

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

We spent 7 hours on this.

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.