Caelan Dailey

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?

Jeremy Wu. I submitted the source code for the 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?

Every few hours. When one of us felt like they had a good idea they wrote code. This worked fine. Whoever wanted to write code would write code.

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

We both worked on it a bunch. I plan on working with them 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.)

It's easier to write a program with a java list because they are a lot easier to manipulate. Efficiency in run time may differ for a java list because the array list may be changed easier compared to the java list and has more flexibility.

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

I expect it to be N because it is not that complicated and does not involve a for loop. It runs 1 time based on the item and the more items it increases by N items.

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 timing is equal to the prediction that I put on question 5 beforehand. The contains method for sets of sizes 100000 to 200000 by steps of 100000 increased by double. The timing techniques show that the big-o behavior is close to N times because of how the code is written. If it had a bunch a for loops it would be slowed down a ton. It matched the prediction.

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 n log n to locate the item because it uses binary search. The worst cases scenario is that it takes n^2 to do what it needs to do.

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

10-15 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.