Michael Cline
u0901489
February 5, 2015

Assignment 3 Analysis Document

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

My programming partner was Nicholas Madsen. Nicholas Madsen submitted the source 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?

My partner and I switched between roles quite often. I would have not preferred to switch roles less, because, I liked both, switching between the driver and the navigator, allowed for me to recuperate from writing code, and the driver allowed for me to be engaged and "learn-by-doing".

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

I will definitely work with Nick again. He has the same work ethic and dedication to meticulous work that I have. He is at the same level of knowledge about programming principles as I am, so we can have brainstorming discussions about solving ideas.

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 the sorted set had used a Java List instead of a basic array, there would be no need to have any of the methods in the sorted set. All of these methods are covered by methods

within the List class that perform the exact same functions. The List has add, append, insert, remove, and toString methods, to name a few, which perform the same functions as the methods we have specified, and they are most likely executed better than our current implementation.
Using a Java List would definitely be more efficient than the basic array implementation. The length of the code for the sorted set would be cut down, because as previously stated, a majority of the functions that were implemented in the sorted set class, are already present in the List class, so there is no reason to specify them again. This means that program development time would be cut in half, if not more, because the Java List would just have to be modified to meet the needs of this specific implementation, which is to be generic, storing any type of element. In terms of running time, the List would be better than the basic array, because the algorithms used to perform functions such as add, remove, or insert, are written efficiently in comparison to the algorithms that were used in this implementation. This translates to a faster running time, because the complexity of the algorithms used in the List class are of an order of complexity lower than our implementation. Referencing the API for the List class, the highest, worst-case complexity in the List class is of order $O(MN)$. This is great in comparison to the array sorted set implementation, which in a worst case has a complexity order $O(N^2)$ for some of the methods.

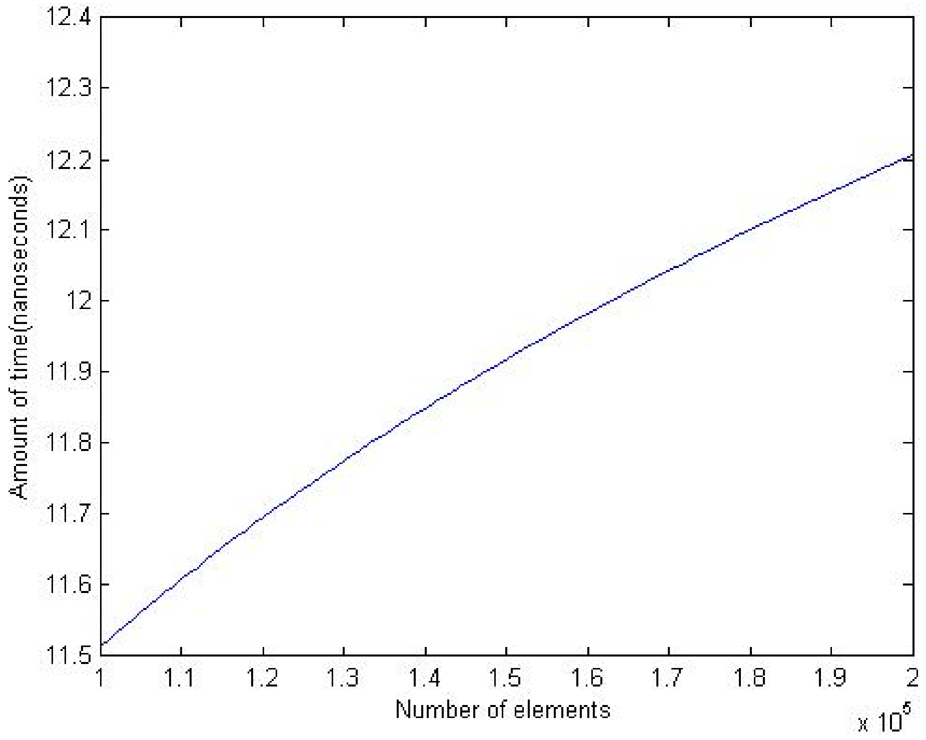
5. What do you expect the Big-O behavior of MySortedSet's contains method to be and why?

MySortedSet's contains method most likely has a Big-O behaviour of $Nlog(N)$, because the methods has to perform one unit of work to run through the if statements to check if there are elements in the array, which is where the N comes from, then the method has to perform a binary search to find the element. We know that the order complexity of

Michael Cline
u0901489
February 5, 2015

binary search is $log(N)$, and finally the method has to return true, which is another unit of work. In the end, collecting all the terms, we end up with the Big-Oh behaviour $Nlog(N)$.

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?



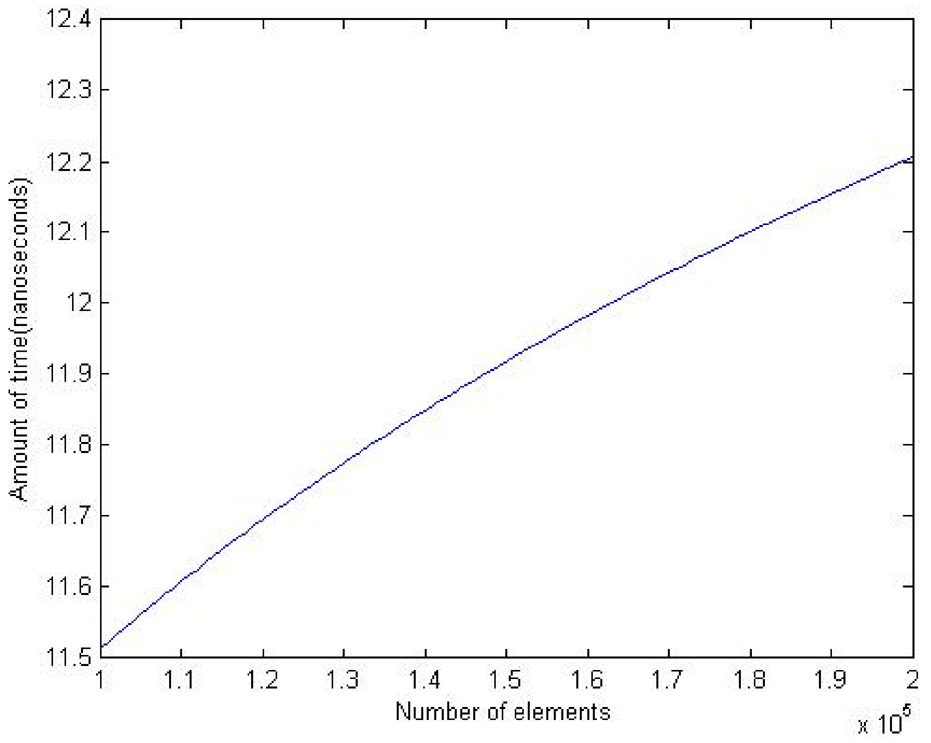
Yes it was as expected.

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

Michael Cline
u0901489
February 5, 2015

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)?

About 175 nanoseconds.



It takes $O(N^2)$.

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

10 hours.