Patrick McHugh / u0883718
CS 2420
Dr. Miriah Meyer
5 February 2015

Assignment 3 Analysis Document

1. My partner is Matt Christensen (u0899251). Matt submitted the assignment for our partnership.

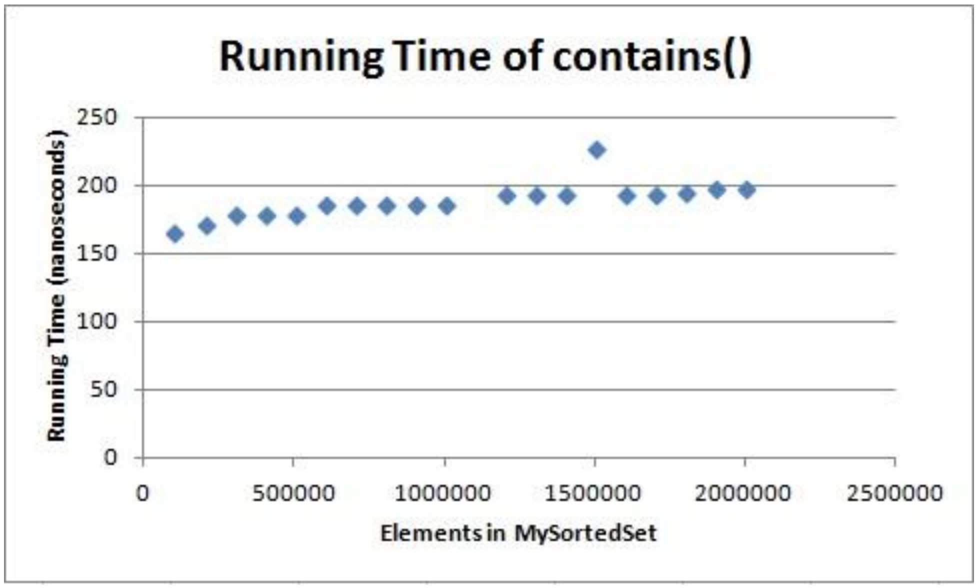
2. We switched roles about every 30 minutes at first, and every 60 minutes by the end, and I preferred switching every 60 minutes. 60 minutes was enough to feel like I was writing substantial portions of code when I was the driver (as opposed to smaller, disconnected pieces), which helped me feel more organized and involved throughout.

3. My partner Matt was great to program with. He is efficient, knows his way around Java, and understood the task at hand. I plan on working with Matt again not only for his skill, but because I believe we work well together.

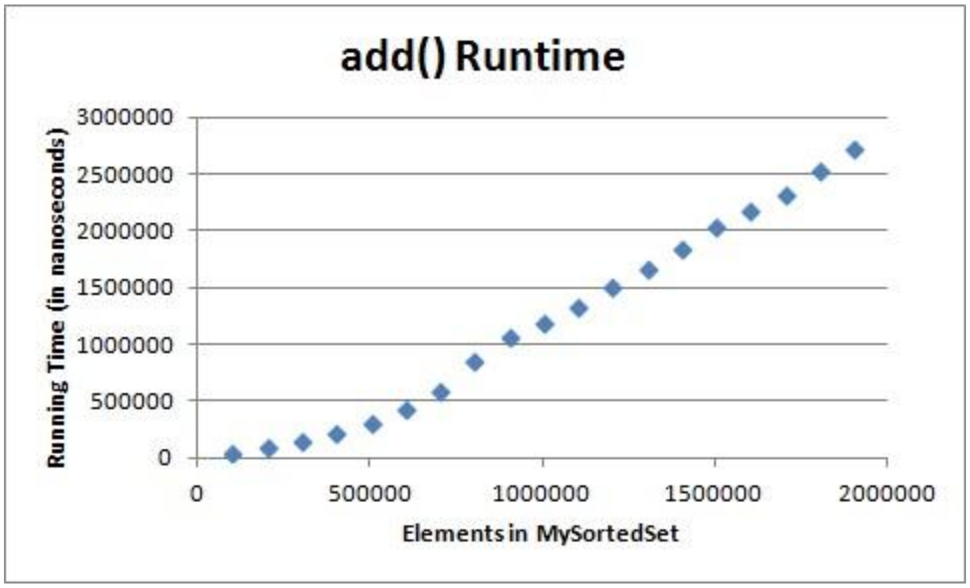
4. Using a Java list, such as an ArrayList, would have made the assignment substantially easier from a programming perspective because many of the methods that were required by the SortedSet<E> interface are already implemented in the ArrayList<E> class, so the methods in MySortedSet<E> would, for some, simply return the result of the equivalent method in the ArrayList<E> class. This would make it less time-consuming to program, but would probably not have much effect on running time, since we essentially implemented the equivalent of an ArrayList.

5. I would expect it to be O(logN), since searching through a set1 that is double the size of set2 would require only one more operation (as is the nature of binary search).

6. This growth rate appears to be about O(logN), which is what I predicted it to be in question 5.



7. I'm going to have to answer this question in two ways, since it's a bit vague. The question asks: "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?" The add method utilizes a method called sortedInsert(), which does a binary search to find the spot where the index should be added, so if you are just asking how long it takes to find the spot where it should be inserted, then the answer is O(logN), but if you are asking how long the method takes to actually add the element at that spot, the answer is O(N), because the method must iterate over each element to the left or right of that spot and move it one over. This analysis is confirmed by the graph below. To the second question: "In the worst-case, how much time does it take to locate the position to add an element (give your answer using Big-oh)?" the answer is O(N).



8. We spent about 8 hours programming, and about 1 hour on the analysis document.