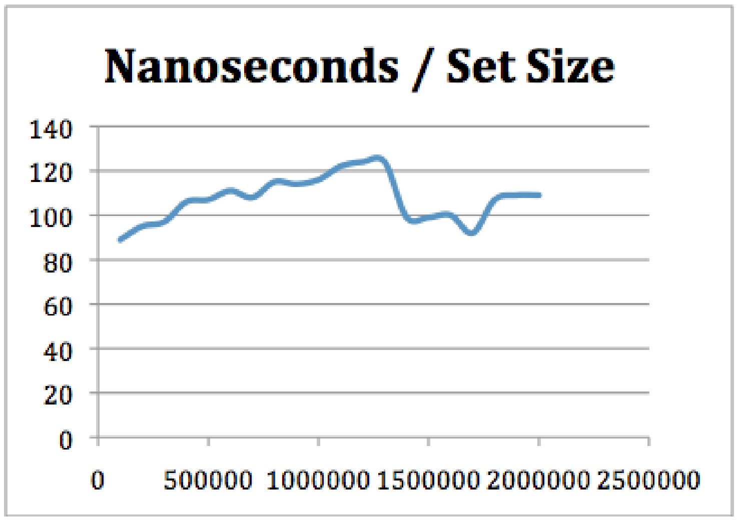
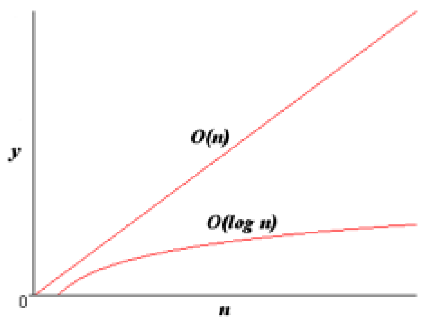
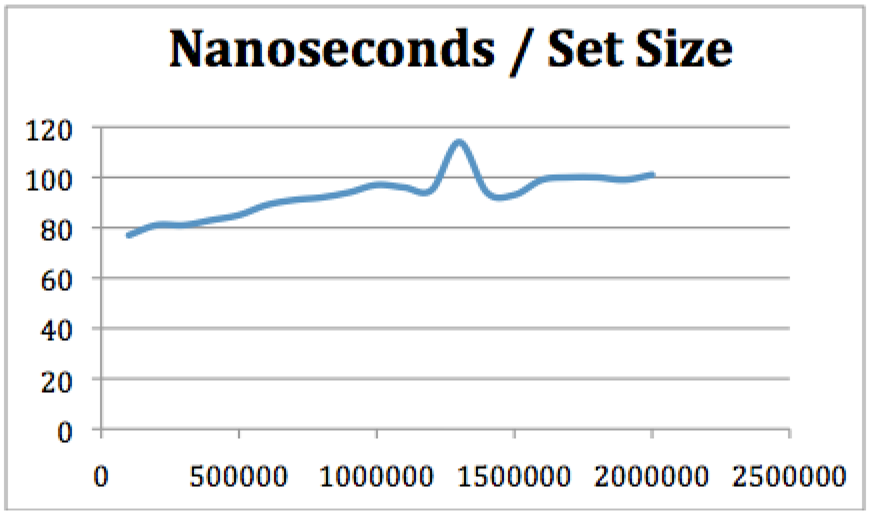
Rachel Brough
U0834533
Assignment 3 Analysis

1. My programming partner was Matthew Brownell. He submitted the source code of our program.
2. My programming partner and I switched off roles every 15 to 20 minutes so we each completed roles for about half the total time. I wouldn't have preferred to switch more or less often; we were both able to successfully complete our roles by switching off within this time frame.
3. My programming partner, Matt, was excellent to work with and strives for equality in our partnership. He is also very intelligent. I would plan to work with him again in the future.
4. If we had backed the sorted set with a Java List instead of a basic array, we would not have to change the size of the array when it fills up. Many of the other implementations would remain similar but using the Java List methods instead of a regular array. I don't believe the running time would differ greatly, but may be slightly faster for basic arrays based on my research. I believe the implementation for a List may have been faster for our development time, but overall I believe these two options are very similar as far as efficiency in creating and running the program.
5. I would expect the Big-O behavior of MySortedSet's contains method to be O(logN), because the contains method refers to the binary search which has a complexity of O(logN).
6. The graph below shows the number of nanoseconds to run MySortedSet's contains method with sizes ranging from 100000 to 2000000. The growth rate in the running times for MySortedSet's contains method corresponds to those predicted in question 5. My data did have a few outliers because of other processes running on my personal computer, but overall the curve is similar to O(logN).





7. The graph below shows the number of nanoseconds to run MySortedSet's add method with sizes ranging from 100000 to 2000000. Again, the data is not perfect because of other process on my computer and the number of times to loop to get the best average. In the worst-case, it takes O(logN) to locate the position to add the element.



8. We spent approximately 8 hours on this assignment.