Yixiong Qin
CS2420
UID: u0900071

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

   Yuhong Lin is my partner during this assignment. And he will submit the source code.

2. What did you learn from your partner? What did your partner learn from you?

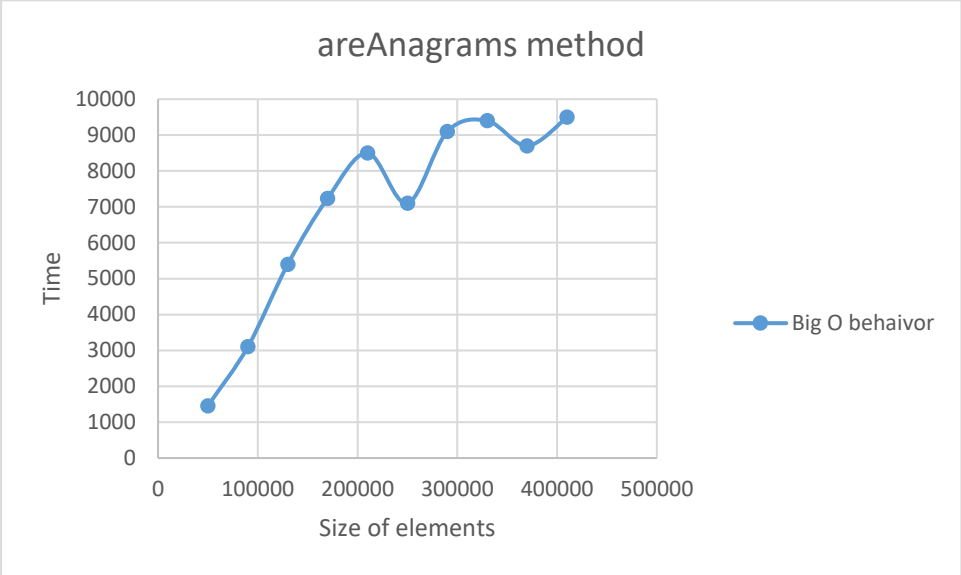
   Sometimes, he can come up with some terrific ideas on logic part of this assignment. And we also took turns to check over each other's code that we wrote. And asked questions or made some suggestions. In addition, I was happy with the way my partner and I worked together. On the other hand, because I had some prior programming experience. Thus, I taught him some of really important concepts of sorting.

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

   We kept helping each other when one of us was getting stuck. We also actively engaging with the task throughout this assignment. I hope I can work with him again.

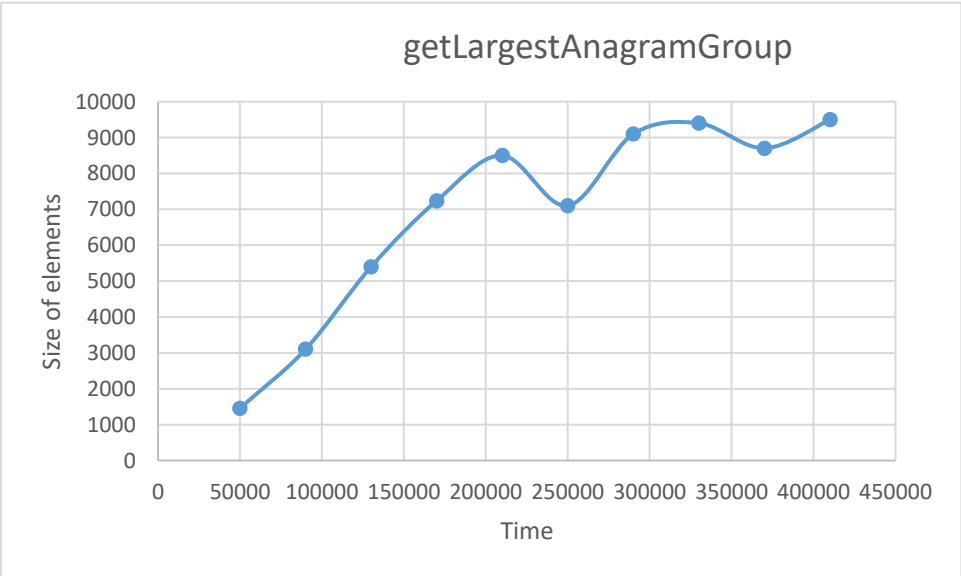
4. Analyze the run-time performance of the areAnagrams method.
-What is the Big-O behavior and why? Be sure to define N.-Plot the running time for various problem sizes (up to you to choose problem sizes that sufficiently analyze the problem). (NOTE: The providedAnagramTester.java contains a method for generating a random string of a certain length.)-Does the growth rate of the plotted running times match the Big-O behavior you predicted?

   The complexity of areAnagrams method is O(N^2). areAnagrams method actually called twice of the for loop on others method. Therefore, we got big O behavior of O(N^2). I implemented 12 tests with increment of 1500. And the graph is roughly as what I expected.



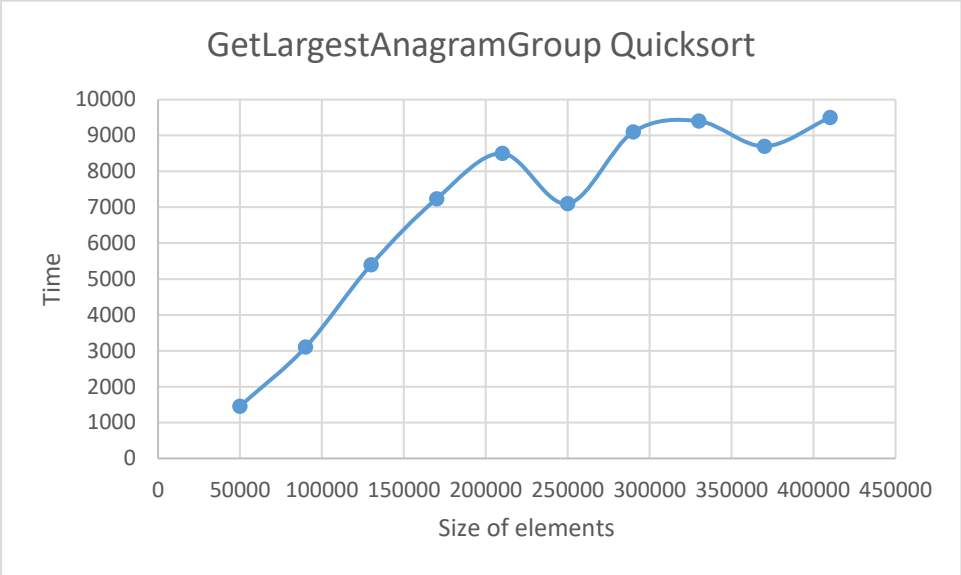
5. Analyze the run-time performance of the getLargestAnagramGroup method using your insertion sort algorithm. (Use the same list of guiding questions as in#4.) Note that in this case, N is the number of words, not the length of words. Finding the largest group of anagrams involves sorting the entire list of words based on some criteria (not the natural ordering). To get varying input size, consider using the very large list of words linked on the assignment page, save it as a file, and take out words as necessary to get different problem sizes, or use a random word generator, provided in AnagramTester.java. If you use the random word generator, use a modest word length, such as 5-15 characters.

   The complexity of getlargestAnagramGroup is O(N). Because we only have one for loop inside this method. And we have the best-case for insertion sort. Thus, the big O behavior for this method is O(N). I also changed the size of array with increment of 1000. But the length of array only changed a small range. But my graph is not as what I expected.



6. What is the run-time performance of the getLargestAnagramGroup method if we use Java's sort method instead(http://docs.oracle.com/javase/6/docs/api/java/util/Arrays.html)? How does it compare to using insertion sort? (Use the same list of guiding questions as in#4.)

   I tried to implement Arrays inside getLargestAnagramGroup method. And it seems like quicksort method. The complexity of quicksort algorithm is O(N logN) except the worst-case. As we can see, it takes less time as the data is getting large. I tested this method with increment of 1000. I could say javas sort method is faster than insertion sort method.



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

   I spent 13 hours on this assignment.