Daniel Zhu

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

Angel Dhungana, I did

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

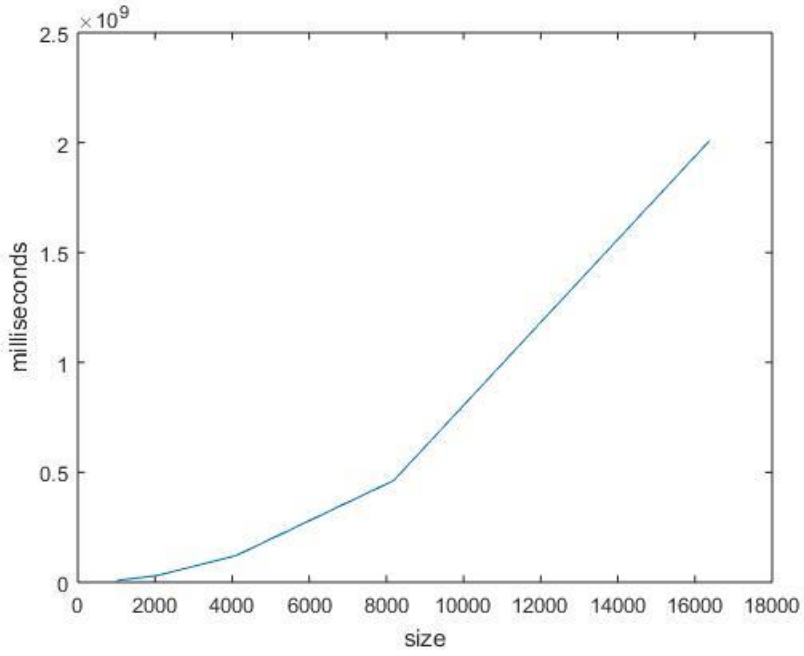
I did not learn much from my partner.

He learned A variety of java functions, how to set up objects, comparators, and how to debug better.

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

He had rather low level skills, and mostly just stared at the screen when it came to writing the code for the program. He wrote the test codes, but mostly did cases he knew about, and would only take notice of edge cases when asked. If possible, I would prefer not to.

4. Analyze the run-time performance of the areAnagrams method.

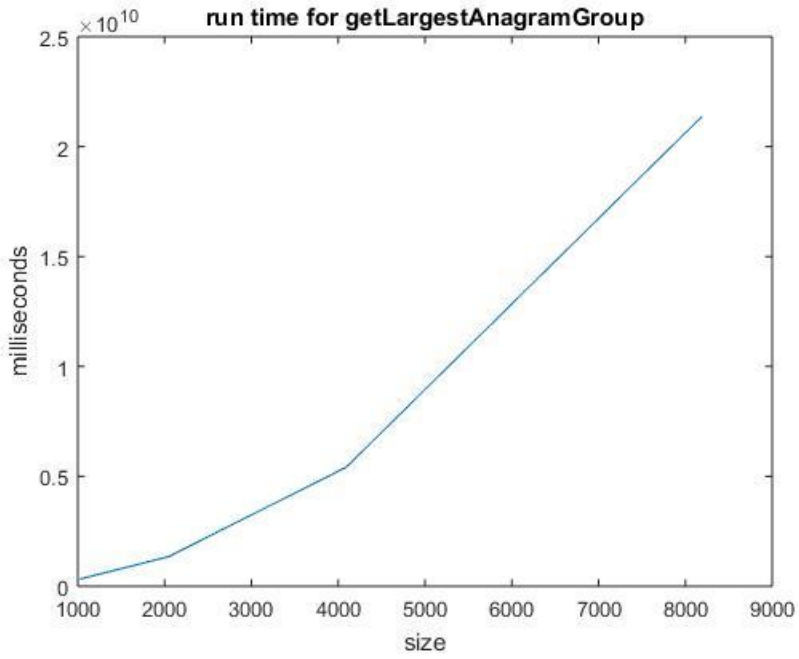



Appears to be O(N^2), due to the upwards curve, this is what was expected. Since an insertion sort was used the time expected was supposed to be O(N^2). The timing took very long to run, so we were only able to get 5 points.

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.

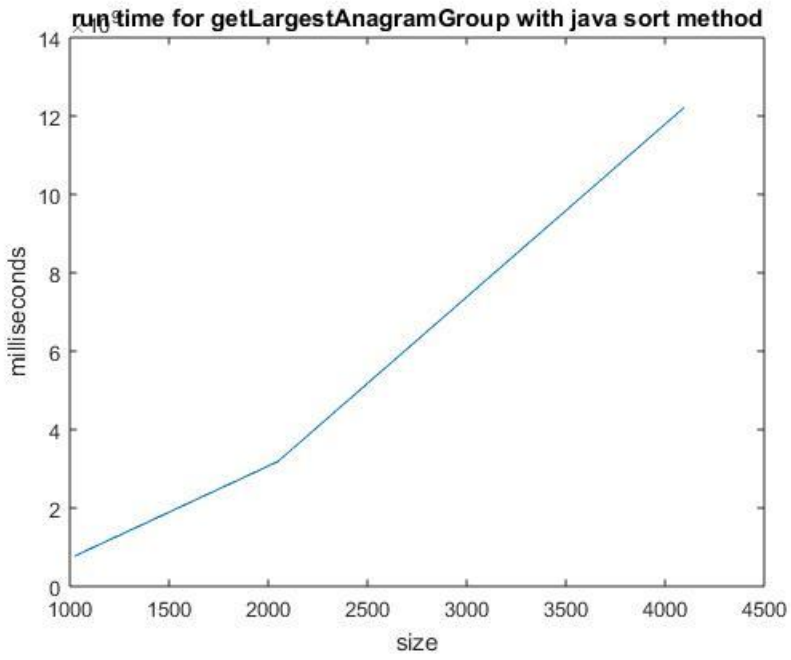
We expected this to be O (N^2), due to the insertion sort and the nested for loop inside the getLargestAnagram method.




The graph resembles the start of an O(N^2), method, which is sadly within our expectations. The timing took very long to run, so we were only able to get 4 points.

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.)

From the data points retrieved, it seems to be the beginning of an O(N^2). It appears to be generally slower than the insertion sort method(or it could have just been that my computer was slower than my partner's computer). The timing took very long to run, so we were only able to get 3 points.




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

10-15 hours