

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

Osama Kergaye is my programming partner. He submitted the source code.

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

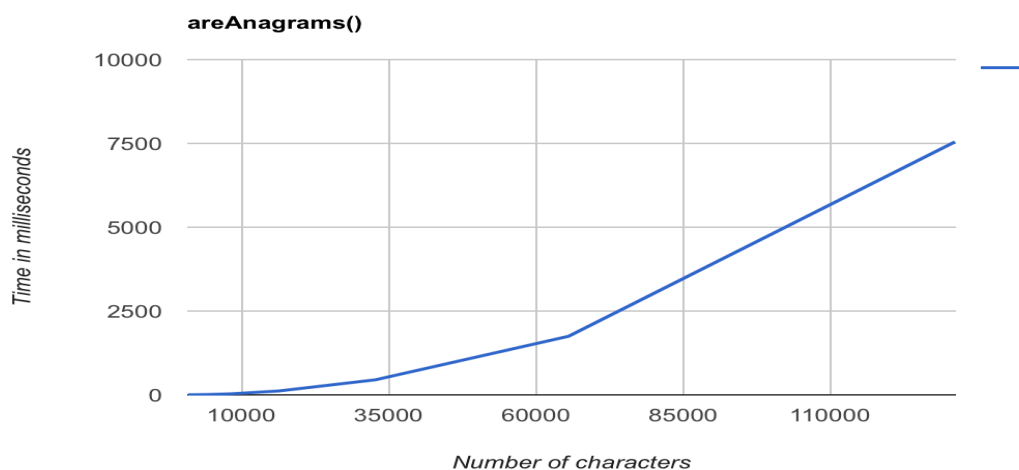
My partner and I have our strengths and weaknesses. I for one am very good at finding bugs within code and how to fix those bugs. My partner on the other hand is very good at writing tests and looking for edge cases. We both learn from each other's strengths.

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

Yes I plan to work with this person again. He is always looking to start and finish the assignment early. He is up to par with my own level of skill and he strives for an A which is important to have in a partner.

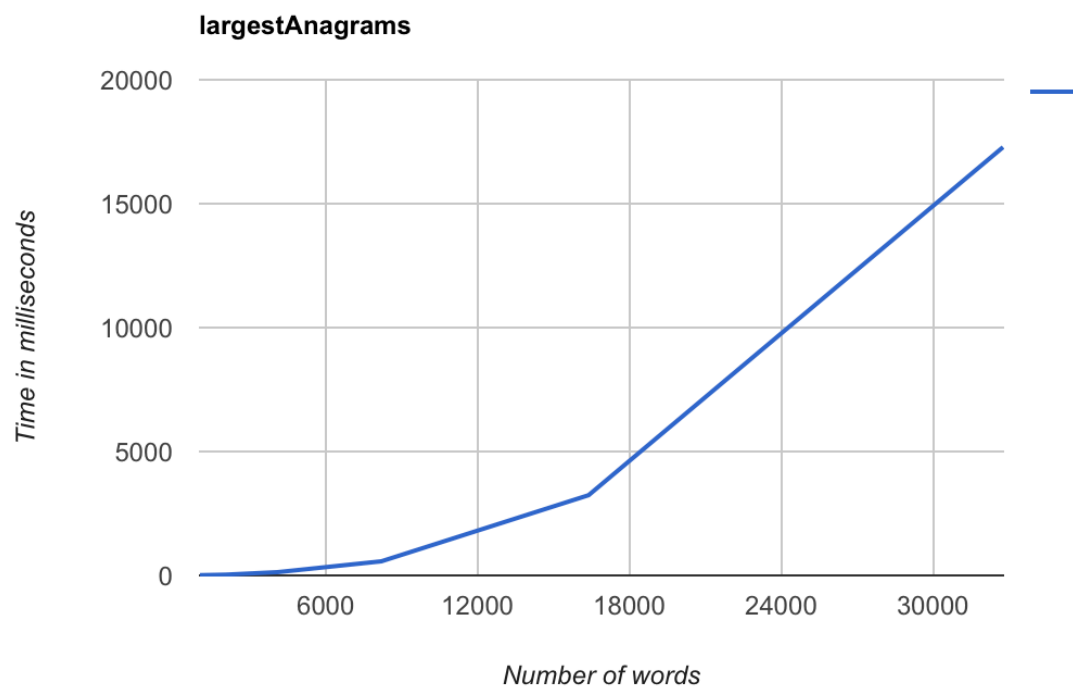
4. Analyze the run-time performance of the areAnagrams method.

The Big-O behavior is N^2 because areAnagrams uses insertion sort and the Big-O behavior of insertion sort is N^2 . Although areAnagrams preforms the insertion sort multiple times ($N^2 + N^2$) the coefficient of N^2 is negligible. The graph matches the running time of the behavior I predicted.



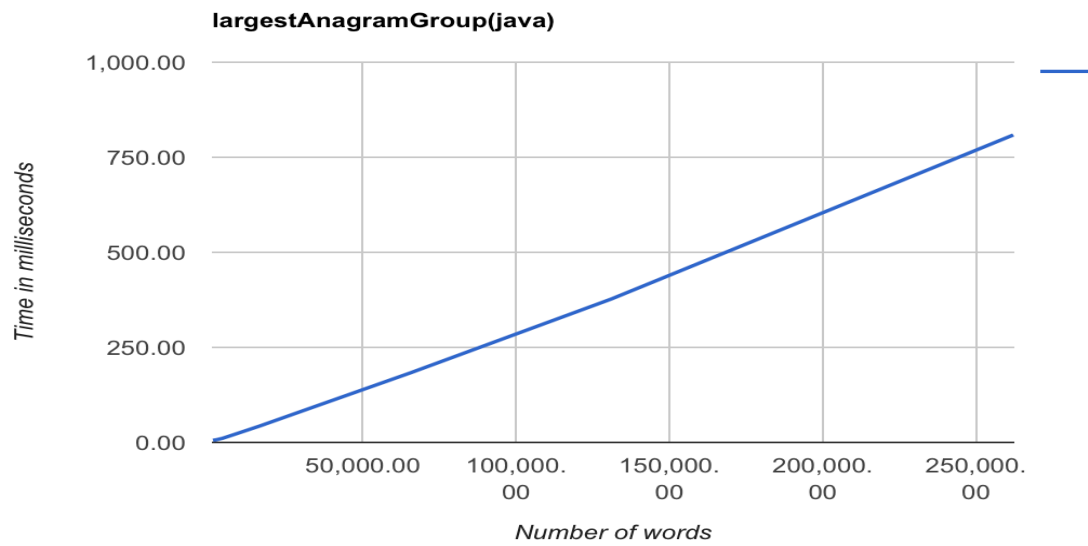
5. Analyze the run-time performance of the `getLargestAnagramGroup` method using your insertion sort algorithm.

The performance of `getLargestAnagramGroup` is N^2 complexity. This is because it is performed using insertion sort. Although the method does have to perform insertion sort multiple times, coefficients are not a concern when it comes to the complexity of the algorithm.



6. What is the run-time performance of the `getLargestAnagramGroup` method if we use Java's sort method instead? How does it compare to using insertion sort?

The big O complexity of Java's `Arrays.sort` method is $N \log N$, this is because Java's method uses a quicksort algorithm. This corresponds to the graph below.



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

I spent around 6-7 hours on this assignment.