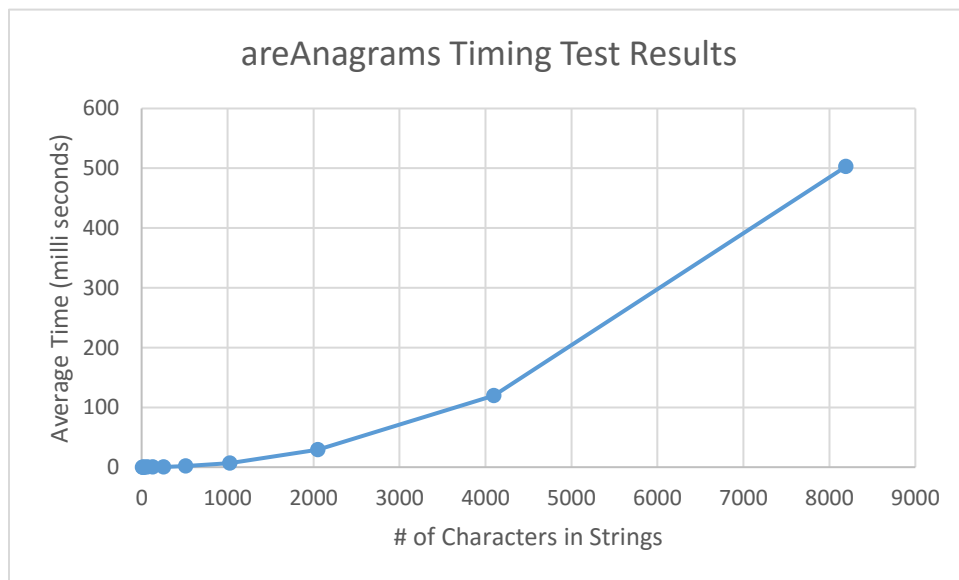


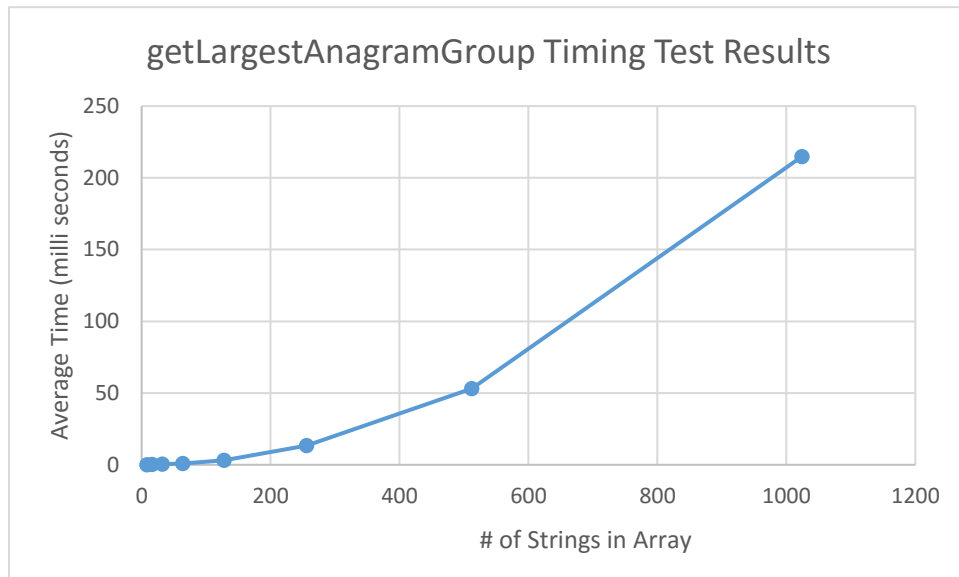
\*This assignment was a special case for me and Kira. I was out of the country for a climbing competition, so we were unable to work together for the majority of the coding. We each wrote code individually, and upon my return we compared code, discussed the problem, and made a solution with the best of both our minds.

1. Who is your programming partner? Which of you submitted the source code of your program?  
Kira Parker. She did.
2. What did you learn from your partner? What did your partner learn from you?  
I learned that sometimes you just have to code first and debug later, especially when one is unsure that one's solution is correct. I like to think that she learned effective commenting. Having to try to read her code without knowing the process by which she wrote it gave us insight into how it feels to try to read poorly documented code from another programmer.
3. Evaluate your programming partner. Do you plan to work with this person again?  
She is very smart and is thus a desirable partner to work with. Yes I do plan to work with her again.
4. Analyze the run-time performance of the areAnagrams method.

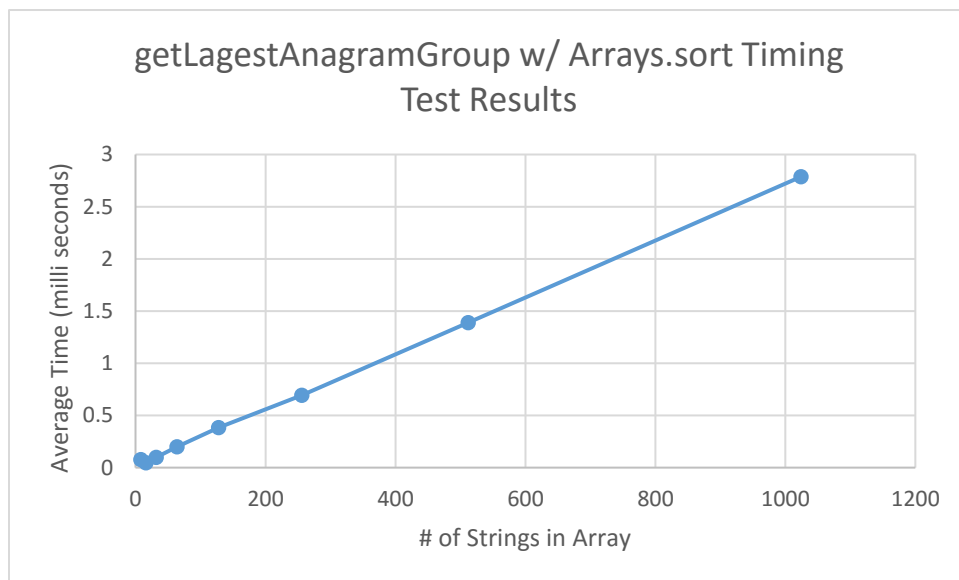


- a. What is the Big-O behavior and why?  
Let  $N$  be the number of characters in both strings being compared. The Big-O behavior is  $N^2$  because of the insertionSort method. With a nested while loop, on average each element is compared to half of the characters sharing its array. Therefore as  $N$  increases, the number of elements being sorted increases as well as the number of items an element must be compared to. The plotted running times support this logic.

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

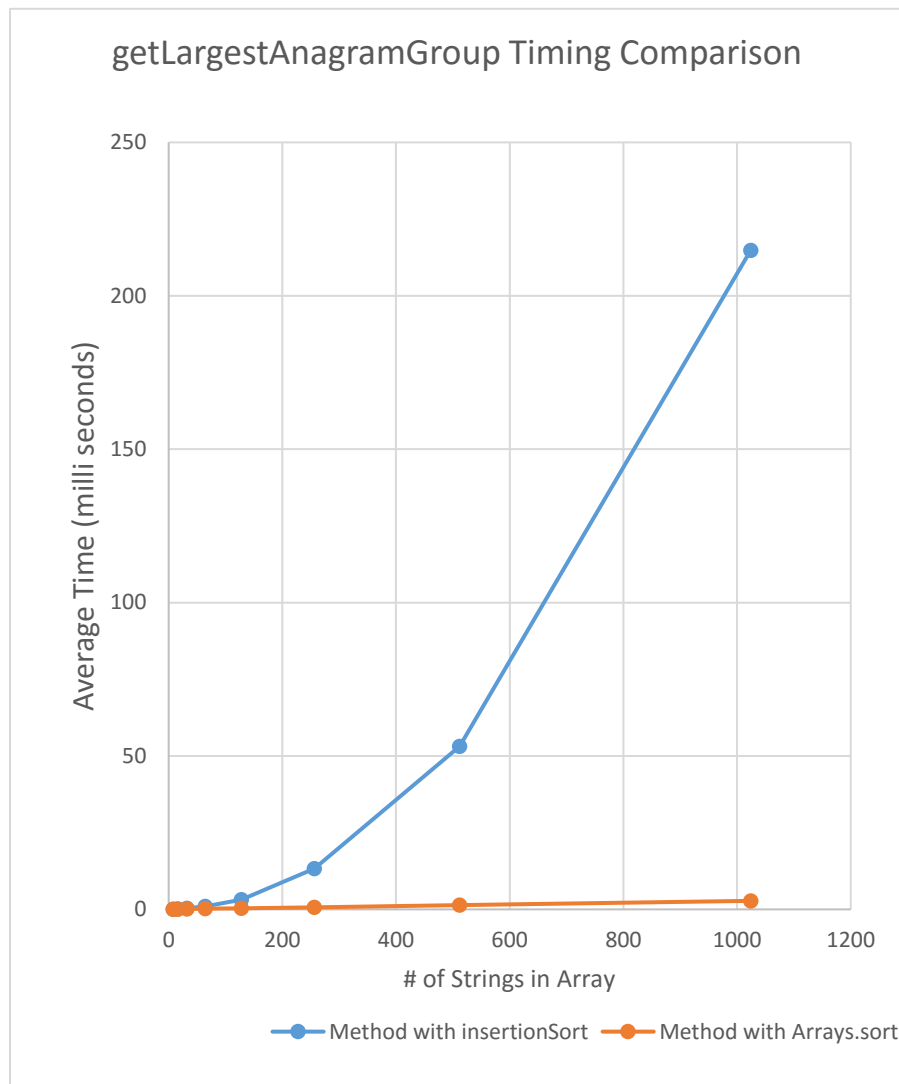


- a. What is the Big-O behavior and why?  
Let  $N$  be the number of strings in the array. The Big-O behavior is also  $N^2$  because, like `areAnagrams`, this method relies on the `insertionSort` method. The plotted running times support this logic.
6. What is the run-time performance of the `getLargestAnagramGroup` method if we use Java's sort method instead?



The Big-O behavior is linear.

How does it compare to using insertion sort? (Use the same list of guiding questions as in #4.)



The difference in time efficiency is a single power of  $N$ . With larger amounts of data, the difference is clear.

7. How many hours did you spend on this assignment?  
7-8.