Nickolas Komarnitsky

U0717854

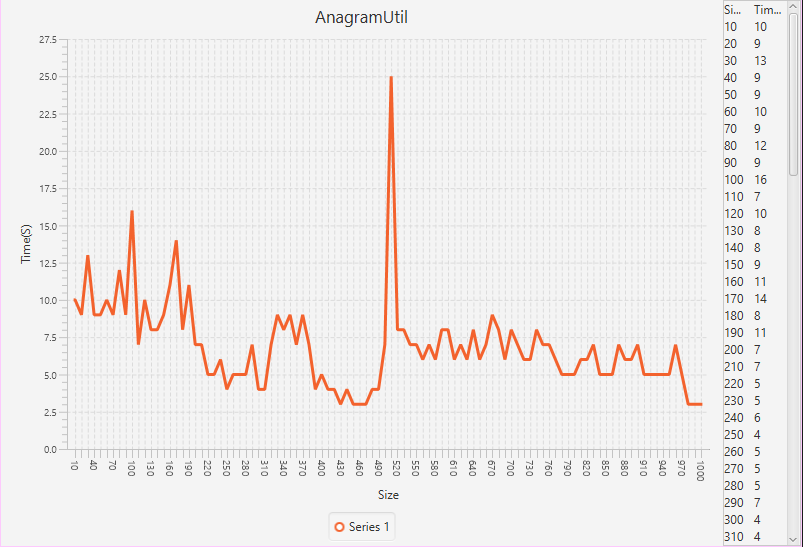
02/06/2017

2402

Assignment04

What is the Big-O behavior and why? Be sure to define N.

Plot the running time for various problem sizes

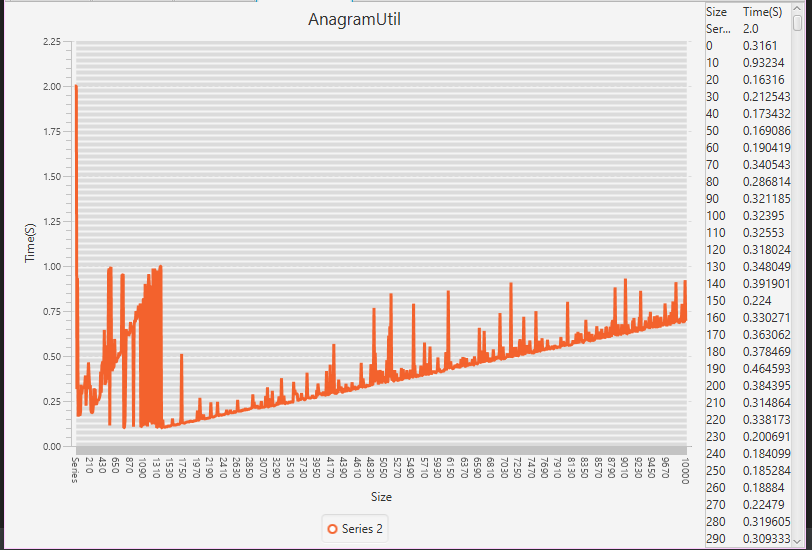


The Big-O behavior is O(N), the run time increases as more and more operations are required. N is the length of the strings. The time required to check the two strings is increased because of the length of the string, but it is almost negligible in the long run.

Does the growth rate of the plotted running times match the Big-O behavior you predicted?

Yes

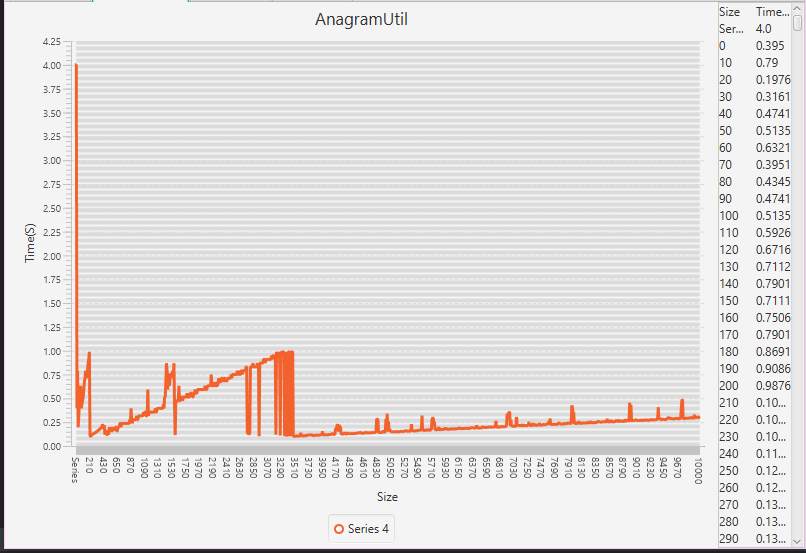
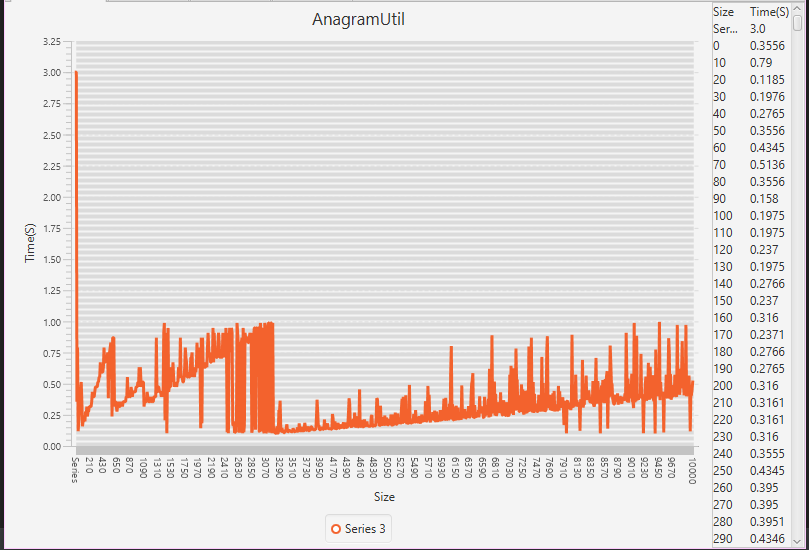
Analyze the run-time performance of the getLargestAnagramGroup method using your insertion sort algorithm. (Use the same list of guiding questions as above.) Note that in this case, N is the number of words, not the length of words. Finding the largest group of anagrams involves sorting the entire list 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 (write a function to randomly build a string of characters). If you use the random word generator, use modest word lengths, such as 5-15 characters.



The get largest anagram group takes only a few seconds, even with an array of 10000 items. It ends up with there being very fast times, but only in some cases. It does best with around 1000 items in the array.

Analyze the run-time performance of the insertionSort method using an array of strings and an array of integers. Does the speed of computing match the speed of the getLargestAnagramGroup? Explain why/why not.

Yes, they match because the insertion sort is the longest part of the algorithm. The has very little effect on the time



What is the run-time performance of the getLargestAnagramGroup method if we use Java's ArrayList sort method instead of our own (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 above.)

It runs a lot faster with the slowest time under 1 second. Using insertion sort takes over 1 second in most cases.

