**Ferdinand Mudjialim**

**Discussion**

By looking at the timing of the different algorithms, one can determine an approximation of the “speed” of the algorithm. Namely, one can even determine the most probable Big-O of each algorithm. First, let’s examine algorithm 1. At n = 1000, the time was about 0.15s, and at n = 2000, the time was about 0.60s. By doubling the input, the time was quadrupled, indicating a Big-O of n^2.

For algorithm 2, at n = 40000, the time was about 0.015s and at n = 80000, the time was about 0.047s. The input was doubled, but the time was neither doubled nor quadrupled, indicating a Big-O of between n and n^2. One possibility (this makes sense with the .sort() used) is that the Big-O of the second algorithm is n(log(n)).

Algorithm 3 was a bit different that the rest in that it took significantly more time than the others. At n = 8, 0.015s was taken, and at n = 10, the time jumped to about 1.6s. The time taken was multiplied by a factor of about 100 while the input was only increased by a factor of 5/4. This data makes sense with the Big-O of n!, as the difference between 8! and 10! is about 9\*10 = 90, close to the multiplier of the time.

Finally, algorithm 4 was the most efficient over larger inputs of n. At first, it seemed that it had a constant Big-O (looking at n = 2000 vs. 10000), but intuitively, that does not make sense with the actual algorithm. Looking at n = 1000, the time taken was about 0.015s while at n = 2000, the time was about 0.030s. This indicates that algorithm 4 has a Big-O of n, as the input vs. time is linear.

**Appendix**

...timing anagramSolution1 with n = 100

True

0.0

...timing anagramSolution2 with n = 100

True

0.0

...timing anagramSolution4 with n = 100

True

0.0

...timing anagramSolution1 with n = 1000

True

0.17189311981201172

0.1406724452972412

0.171891450881958

...timing anagramSolution2 with n = 1000

True

0.0

...timing anagramSolution4 with n = 1000

True

0.0

...timing anagramSolution1 with n = 2000

True

0.7501158714294434

0.562558650970459

0.5313048362731934

...timing anagramSolution2 with n = 2000

True

0.0

...timing anagramSolution4 with n = 2000

True

0.015587806701660156

0.015627384185791016

...timing anagramSolution1 with n = 10000

True

17.486906051635742

13.329568147659302

13.345192909240723

...timing anagramSolution2 with n = 10000

True

0.0

...timing anagramSolution4 with n = 10000

True

0.01563096046447754

0.01562952995300293

0.015627145767211914

...timing anagramSolution1 with n = 20000

True

71.6972770690918

56.22537565231323

54.41121459007263

...timing anagramSolution2 with n = 20000

True

0.015631914138793945

0.015586137771606445

0.01558685302734375

...timing anagramSolution4 with n = 20000

True

0.031247615814208984

0.035663623809814453

0.035660762786865234

...timing anagramSolution2 with n = 40000

True

0.015440702438354492

0.015484094619750977

...timing anagramSolution2 with n = 80000

True

0.046724796295166016

0.04673933982849121

0.046738624572753906

...timing anagramSolution3 with n = 5

True

0.0

...timing anagramSolution3 with n = 8

False

0.015627384185791016

0.015626907348632812

0.015589475631713867

...timing anagramSolution3 with n = 10

True

1.6407577991485596

1.5470390319824219

1.5938866138458252