CS201 Homework 2

Sec 1

Kemal Sarper Şahin

22103801

Describing the complexity of the algorithms

Algorithm 1 is O(n^2) because in first loop it will look through n integers, then n-1 and so on till n/2. So mathematically it is n + n-1 + n-2+…..+n/2 ,therefore it will take n*(n-1)/2 - ((n/2) * ((n/2)-1)/2) which is O(n^2).

Algorithm 2 is O(nlog(n)) because the algorithm is quick sort + 1. Quick sort is nlog(n) because in every recursive depth the algorithm iterates through a total of n integers, and total depth is log(n). Therefore n * nlog(n) + 1 which is O(nlog(n)).

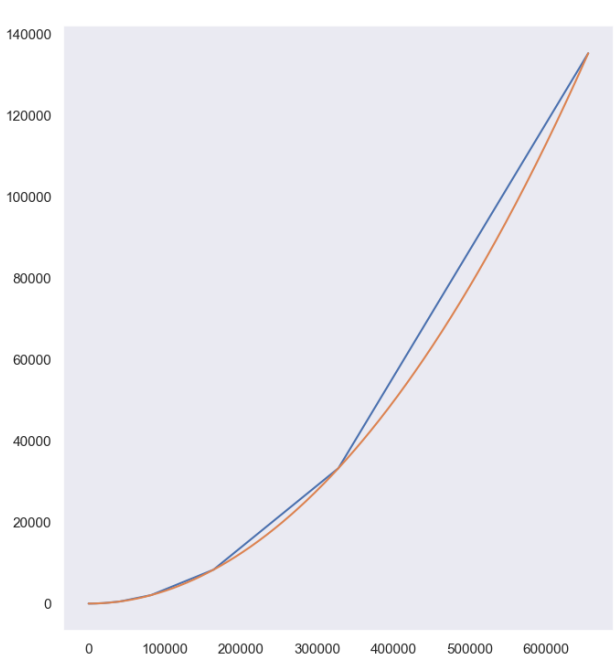
Algorithm 3 is O(n). Algorithm 3 takes the array and splits it into 2 smaller problems, then takes one of these and splits it into another 2 problems and so on. This gives us a best case of O(n) time complexity.

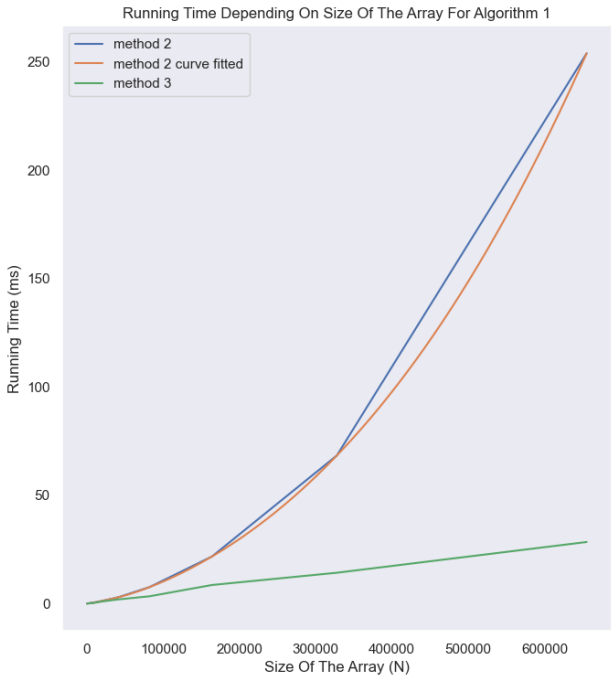
The table and the graphs

I ran every method 10 times and put their average on the table and the graphs.

n/method	m1	m2	m3
5	0.0001	0.0002	0.0002
10	0.0002	0.0003	0.0012
20	0.0004	0.0007	0.0028
40	0.0021	0.0021	0.0031
80	0.0027	0.0031	0.0056
160	0.0096	0.0064	0.0089
320	0.0354	0.0131	0.0189
640	0.1312	0.0281	0.0299
1280	0.5189	0.0562	0.0761
2560	1.98	0.1331	0.1223
5120	7.99	0.2665	0.2245
10240	31.93	0.55	0.39
20480	129.08	1.27	1.01
40960	513.53	2.91	1.92
81920	2079.93	7.60	3.37
163840	8335.39	21.74	8.58
327680	33308.80	68.39	14.24

655360	135399.00	254.09	28.43





Discussion

My obtained results shows the growth rates clearly. It is easy to see that algorithm 1 is quadratic, 2 is nlog(n), and 3 is n. I specifically chose my n's as doubling with each n so that times could be compared easier. The values are really close to the interpolated values. Algorithm 1 is expected to have a quadratic form, and it is really consistent with theoretical values. Algorithm 2 seems like the results are suitable with expected results. Same thing could be said about the 3rd algorithm.

PC specs

Operating System : Microsoft Windows 10 Pro

CPU : 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz, 2304 Mhz

RAM : 16 GB