Homework Assignment 2: Algorithm Complexity Analysis

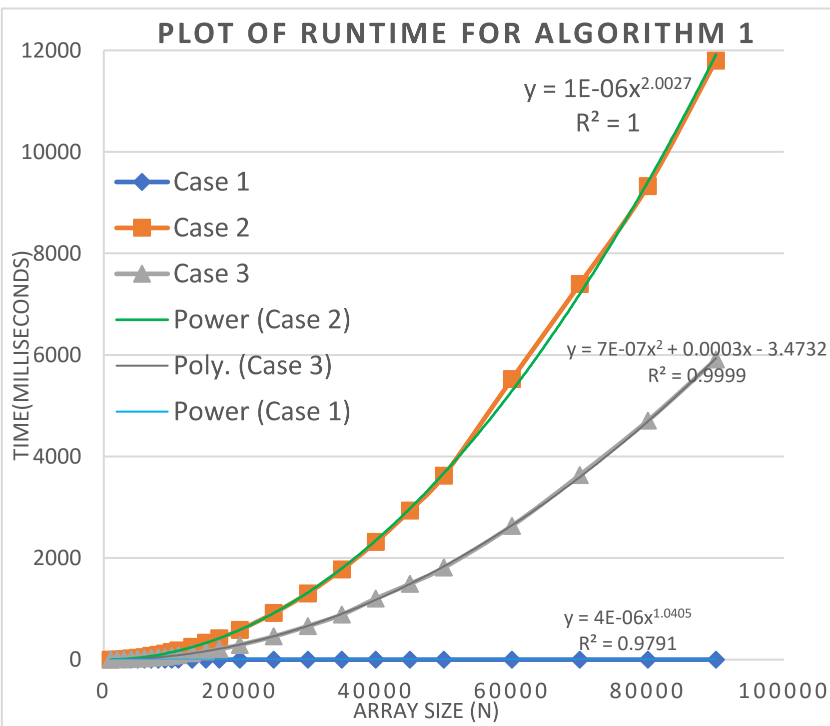
Name: Mert
Surname: Ertuğrul
ID: 21703957
Section:
Date: 28.11.2020

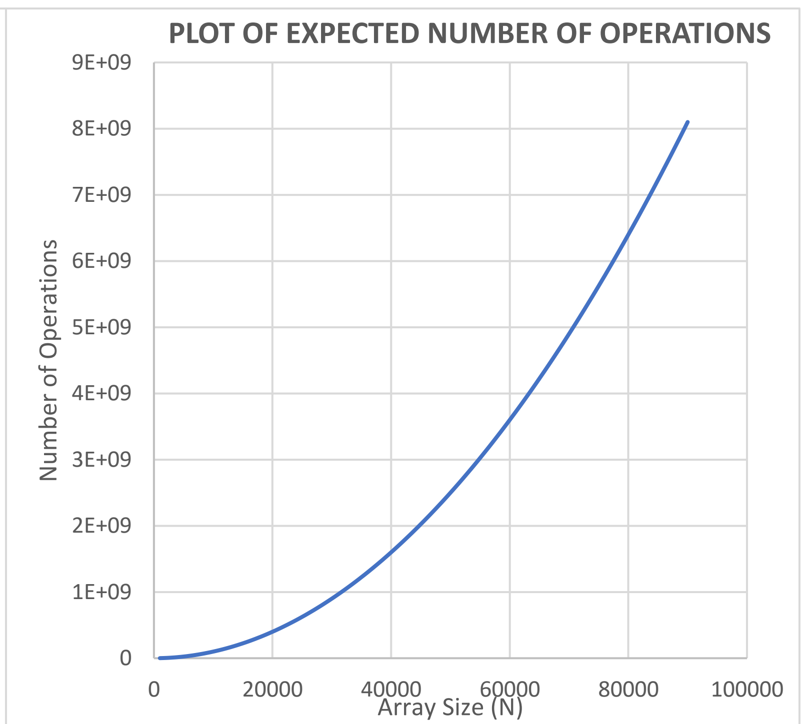
Results for Algorithm 1

The datapoints obtained for each array initialization ordering case are given separately for algorithm one in the runtime vs array size plot below. The original datapoints are given in the appendix. Arrays were initialized using random numbers. Several iterations of a function call had to be measured together at a time to avoid getting zero millisecond computation time and the result was divided by iteration number.

It appears that for the first algorithm, the worst case is given by array ordering case 2, while case 3 (no specific ordering) can be labelled as the average case. The best case scenario is case 1. When we add a trendline to the worst-case result (case 2), we see that a power series with approximately x^2 fits well to the data. Therefore, the empirical worst case is $O(n^2)$. The empirical best case appears as if it is O(1) but that is only because of the scaling of the plot, in fact it is O(n), as no shifting of elements will be needed for case 1. In the average case, about half of all elements in array two would trigger a shift. The average case is still in quadratic time but with a lower constant coefficient.

On the other hand, the expected worst case is also $O(n^2)$, since in the worst case, the elements of array one that were already placed in array three will all have to be shifted for every element of array two. If every element of array two is smaller than those of array one, the N elements of array one are shifted for each of the N elements of array two. This is caused by the requirement that we should start merging array two in from its beginning and not its end. If we were to begin from the end, time complexity could be lower. We have $O(n^2)$, where n^2 from the shifting dominates the number of operations and linear terms are neglected.

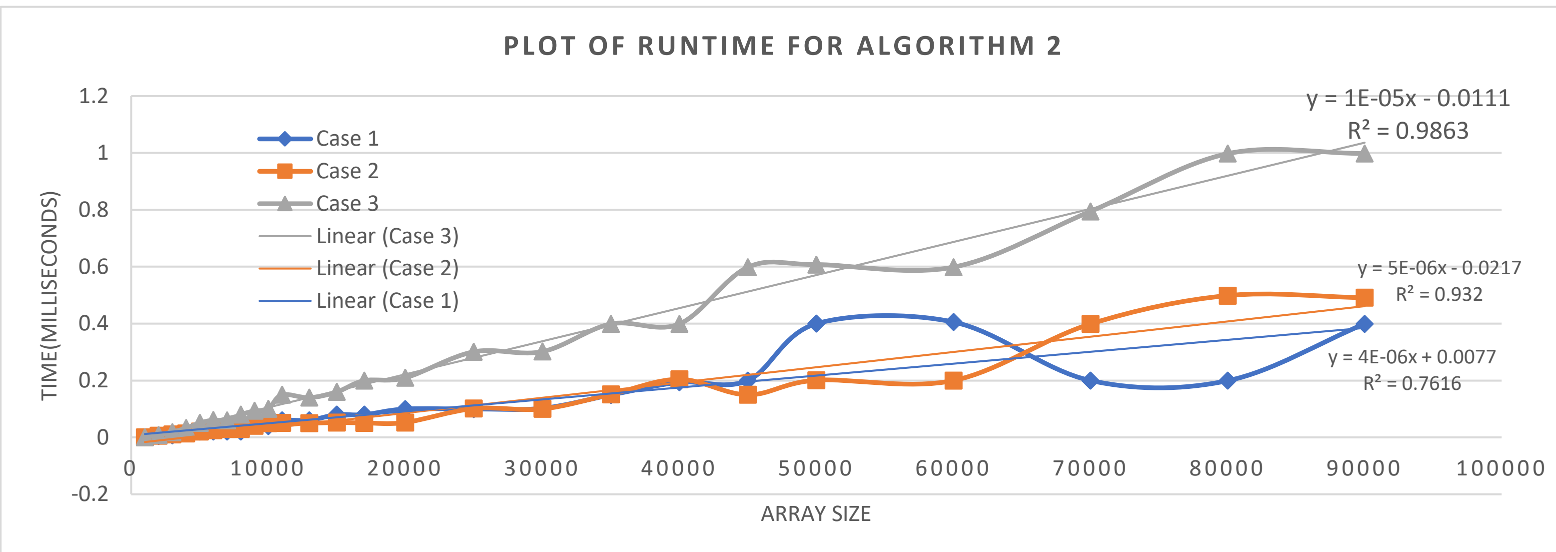




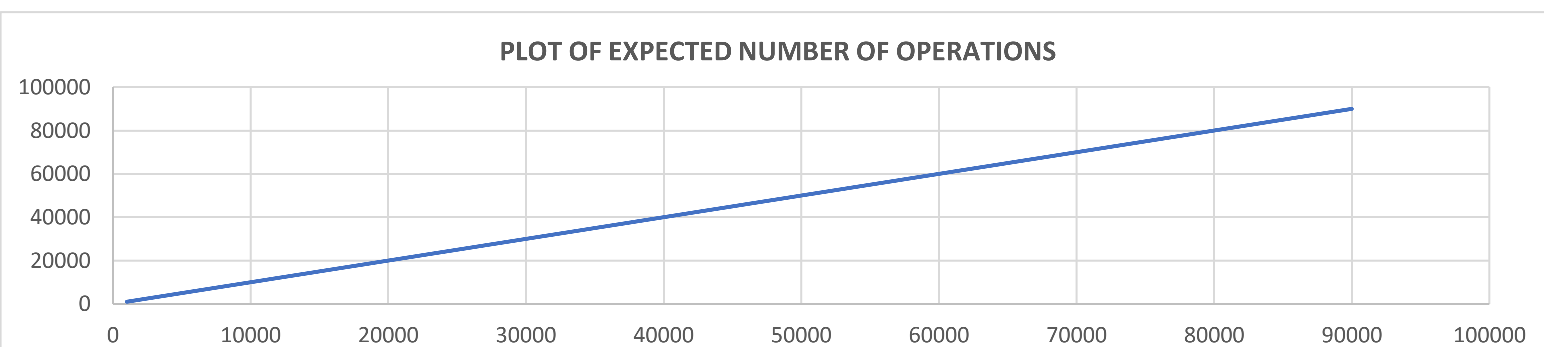
Results for Algorithm 2

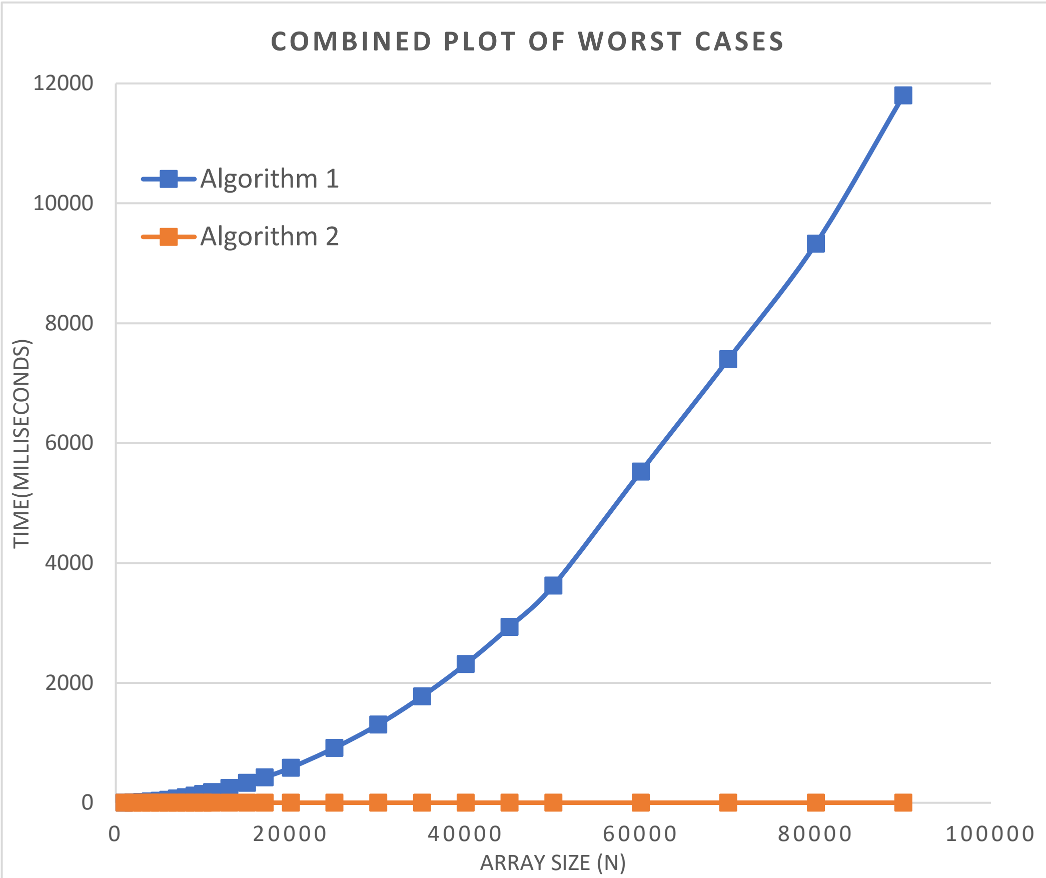
The datapoints obtained for each array initialization ordering case are given separately for algorithm two in the runtime vs array size plot below. The original datapoints are given in the appendix.

For the second algorithm, the worst case is given by array ordering case 1(no specific ordering of arrays), while case 2 and 3 show entangled results. Whether the first or second array has smaller elements than the other would not change the computation time as the algorithm is symmetric with respect to array one and two. Therefore, this entanglement is to be expected. When we add a trendline to the worst-case result (case 3), we see that a linear equation with $10^{-5}x$ fits well to the data. Therefore, the empirical worst case is O(n). The best case could be both case 1 and 2 as their results average out to be approximately the same linear trend line. The average case would be somewhere between case 1 and the pair of case 2-3, another linear pattern with a slope smaller than 10^{-5}. Best, average and worst cases are all O(n) since no nested loops are present and each element can be traversed once. In the worst case, the difference in coefficient is caused by the if-else statement being used more than N times unlike in cases 1 and 2.



The expected worst case is also O(n) as the elements of array one and two are each traversed only once. In the absolute worst case, the last element of each of the two arrays should be such that they will fall side by side in array three. Then, the number of if-else statements would be maximized at 2N-1 which would change the coefficient of the linear term. The expected number of operations is given as f(N)=N without any coefficient to represent the big-Oh notation. In reality, each addition, assignment, test operation would come together to give a coefficient.





Overall, the expected growth rates and empirical growth rates have been on the same asymptotic order. The first algorithm was observed to be in the order of $O(n^2)$ while the second one in the order of O(n). Trendlines were added to the results to prove the validity of the comments made. The data points for cases 1 and 2 have large jumps and irregularities that decrease the confidence of the trendlines. In theory, they are expected to be linear and the data confirms this to a certain degree but the fact that their time scales are comparatively smaller and external influences on computation speed may be the cause of the irregularities.

Computer Specifications

- **Dell XPS 15**
- **OS:** Windows 10 Pro, 64-bit
- **Processor:** Intel Core i7-8750H CPU @ 2.20GHz 2.21GHz
- **RAM:** 16 GB
- **Graphics:** NVIDIA GeForce GTX 1050 Ti Max-Q

Appendix

Table 1: Data Points for Algorithm 1

N	Case 1	Case 2	Case 3
1000	0.004985	1.47836	0.7544
2000	0.00997	5.79758	2.932
3000	0.02002	13.005	6.562
4000	0.01996	23.1158	11.7
5000	0.0203	36.4704	18.18
6000	0.03972	52.2323	26.16
7000	0.03938	71.7736	35.56
8000	0.06072	92.4339	46.49
9000	0.05984	119.05	58.67
10000	0.06074	147.706	73.42
11000	0.07916	179.371	87.32
13000	0.07978	248.183	122.8
15000	0.07876	331.26	165.7
17000	0.0997	422.708	207.1
20000	0.1496	585.349	291.1
25000	0.14965	916.332	457
30000	0.1496	1305.42	661
35000	0.1992	1775.87	886.7
40000	0.2056	2316.59	1201
45000	0.202	2935.32	1492
50000	0.3988	3625.25	1811
60000	0.399	5526.56	2635
70000	0.6066	7401.02	3636
80000	0.3988	9329.44	4708
90000	0.5982	11800	5910

Table 2: Data Points for Algorithm 2

N	Case 1	Case 2	Case 3
1000	0.004982	0.00598	0.008078
2000	0.007482	0.009968	0.01795
3000	0.01994	0.01396	0.03255
4000	0.01998	0.02001	0.05088
5000	0.02072	0.02502	0.06056
6000	0.02	0.0299	0.05992
7000	0.01998	0.02964	0.08013
8000	0.05984	0.04082	0.09334
9000	0.03994	0.0498	0.0999
10000	0.05978	0.0505	0.1496
11000	0.06046	0.04985	0.1397
13000	0.07976	0.0523	0.1596
15000	0.07982	0.0511	0.1995
17000	0.09975	0.0525	0.2096
20000	0.0998	0.1014	0.301
25000	0.1022	0.0999	0.3015
30000	0.1492	0.1512	0.3989
35000	0.1936	0.2043	0.3988
40000	0.1998	0.1497	0.5982
45000	0.3998	0.2011	0.6076
50000	0.4056	0.1998	0.5986
60000	0.1998	0.3988	0.7946
70000	0.1992	0.4985	0.9975
80000	0.3992	0.491	0.998
90000	0.3992	0.3992	0.9954