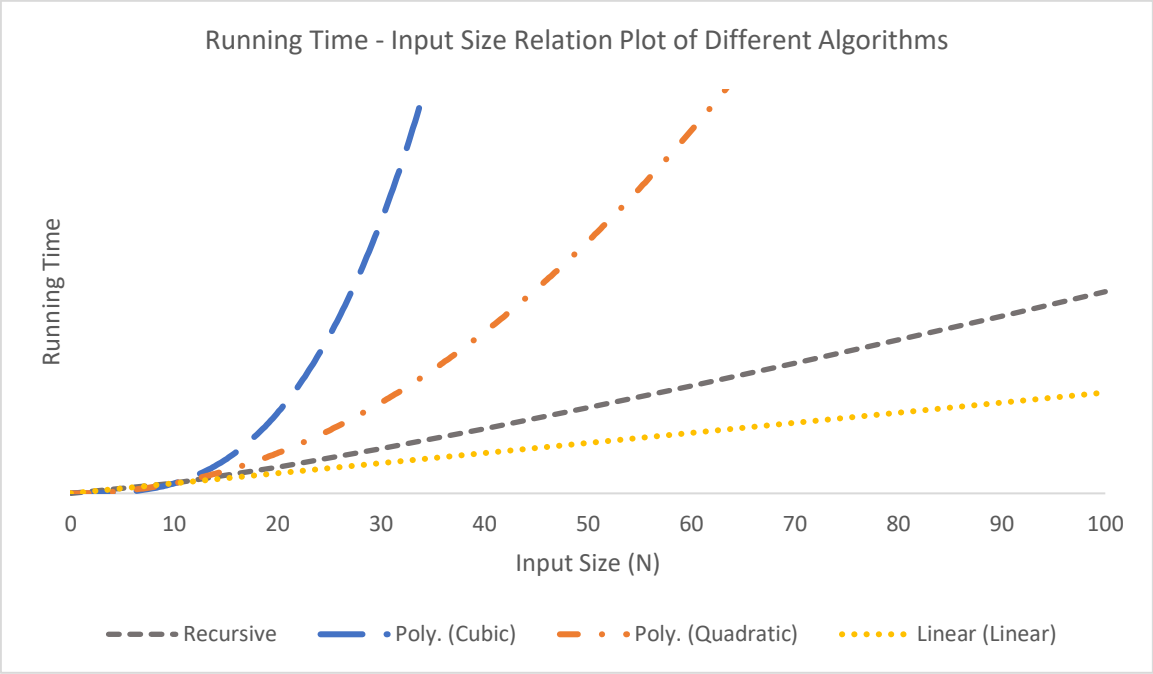
Name: Muhammet Abdullah KOÇ ID: 21802832 CS201 Section 1 Homework 2

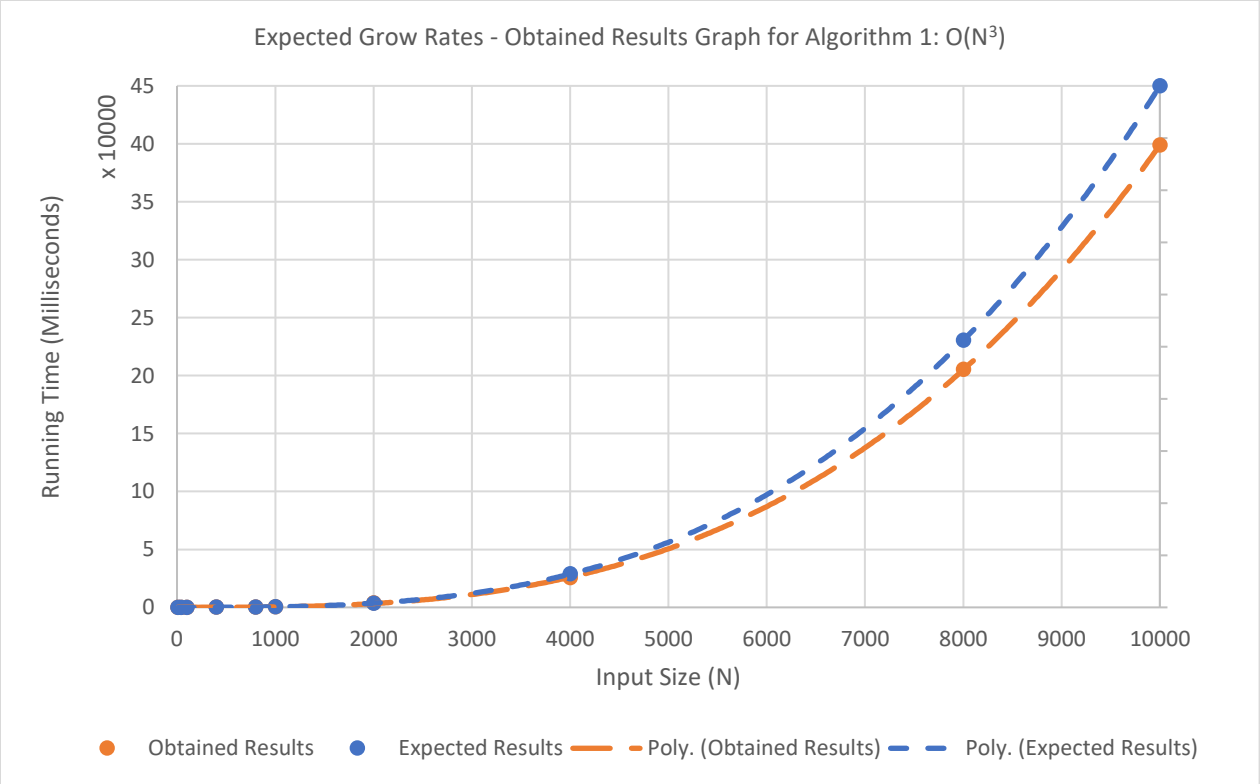
Comparison Table of Algorithms

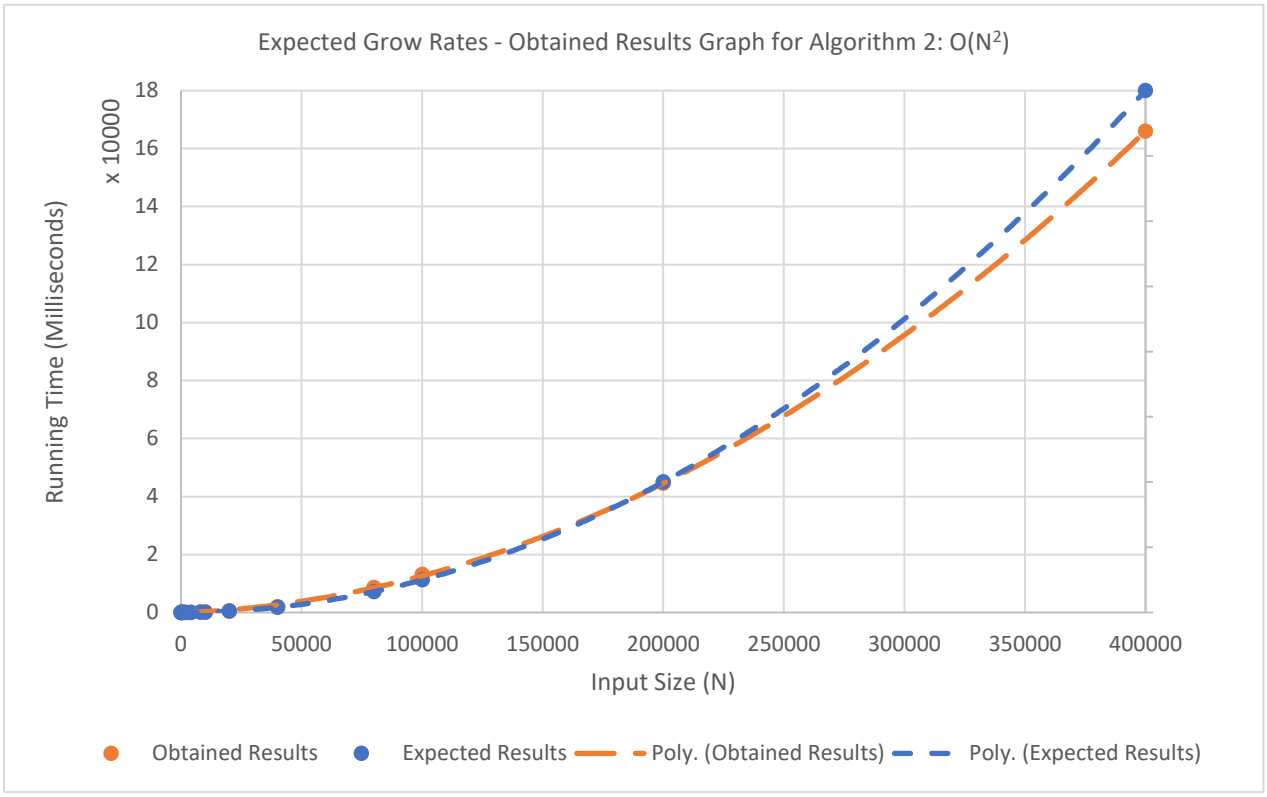
Input Size (N)	Functions – Execution Times (milliseconds)			
	Function 1 - $O(N^3)$	Function 2 - $O(N^2)$	Function 3 - O(N logN)	Function 4 – O(N)
10	0.001	0.0002	0.0002	0.0001
40	0.01	0.0024	0.001	0.0001
100	0.495	0.015	0.003	0.0003
400	36.938	0.21	0.017	0.0008
800	212.517	1.097	0.044	0.002
1,000	417.002	1.566	0.078	0.002
2,000	3,738.85	5.38	0.145	0.006
4,000	25,626.8	21.65	0.345	0.014
8,000	205,213	84.771	0.897	0.033
10,000	398,779	154.556	1.196	0.047
20,000	NA	506.648	1.978	0.086
40,000	NA	2,032.66	4.987	0.175
80,000	NA	8,581.64	10.006	0.353
100,000	NA	13,091.1	12.964	0.473
200,000	NA	44,536	20.965	1.003
400,000	NA	165,961	46.897	1.942
800,000	NA	NA	84.655	3.887
1,000,000	NA	NA	122.531	6.505
4,000,000	NA	NA	484.657	21.674
8,000,000	NA	NA	951.207	44.594
10,000,000	NA	NA	1,251.36	55.269
40,000,000	NA	NA	5,371.69	184.159
100,000,000	NA	NA	13,358.1	467.286

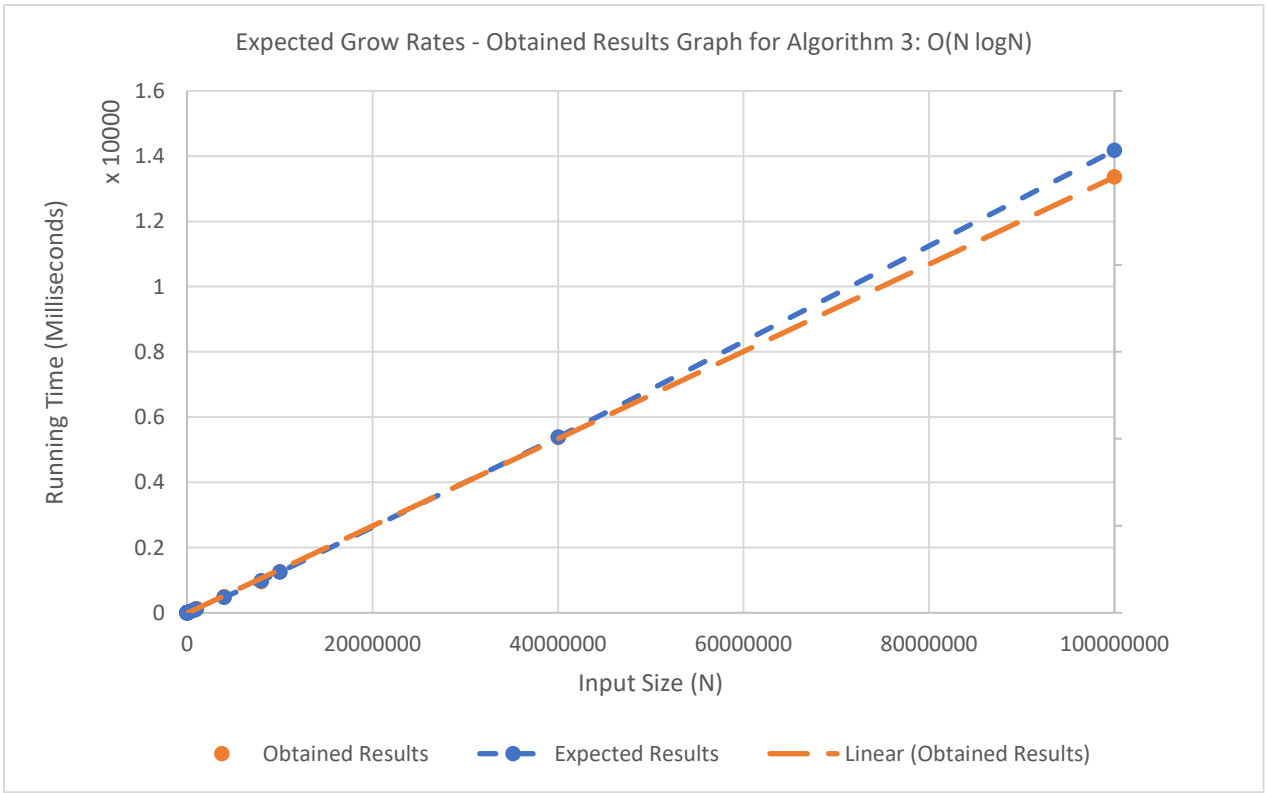
Running Time - Input Size Relation Plot of Different Algorithms

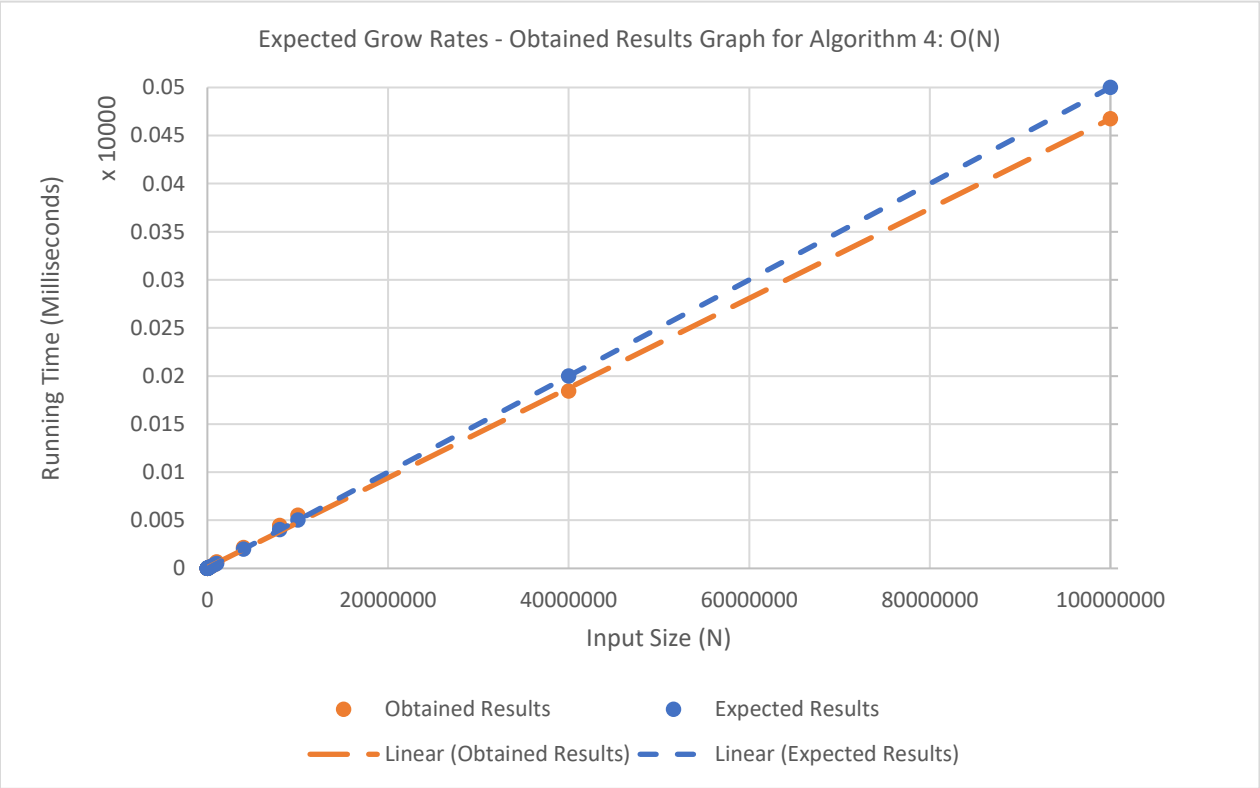


Expected Grow Rate – Obtained Results Graphs for 4 Algorithms




Expected Grow Rates - Obtained Results Graph for Algorithm 2: O(N²)
Running Time (Milliseconds)
x 10000
18
16
14
12
10
8
6
4
2
0
0
50000
100000
150000
200000
250000
300000
350000
400000
Input Size (N)
Obtained Results
Expected Results
Poly. (Obtained Results)
Poly. (Expected Results)


Expected Grow Rates - Obtained Results Graph for Algorithm 3: O(N logN)
Running Time (Milliseconds)
x 10000
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
20000000
40000000
60000000
80000000
100000000
Input Size (N)
Obtained Results
Expected Results
Linear (Obtained Results)



Results

Results show that obtained results are consistent with expected results. Generally, obtained results took less time than expected results. This may stem from computer or environment. In the first algorithm, because algorithm is not efficient, I cannot run the program above 10000 inputs. Below 10000 inputs, program worked as expected. For example, when input size increases from 1000 to 2000, there should be $(2000/1000)^3 = 8$ times difference. Obtained time for 1000 input = 417.002 milliseconds, and 2000 input = 3738.85 milliseconds. 3738.85/417.002 = 8.96. There may be some errors because background services affect the working performance of computer, but the pattern generally matches with expected results.

Patterns are also matching in other three algorithms. However, in all of algorithms, we can see that algorithms works faster compared to expected results as input size increases. In small inputs, obtained results took more time than expected. The reason of this situation may be calculation errors in small inputs. When input size is smaller, running algorithm one time is not enough to calculate elapsed time. Therefore, I used a for loop and divide the result with loop count, and this calculation is not precise.

As a result, elapsed time calculations in my computer are consistent with expected results except some errors that may stem from computer, environment or necessarily used for loops. In addition, effectiveness of algorithms can be sorted as $O(N) > O(N \log N) > O(N^2) > (N^3)$. If it is possible, linear algorithms should be used for problems.

Computer Specifications

MSI Prestige 14 A10SC

Operating System: Windows 10

Processor: Intel Core i7 10710U

Memory: 16 GB DDR3

Graphics: Nvidia GTX1650 Max-Q

Storage: 512 GB SSD