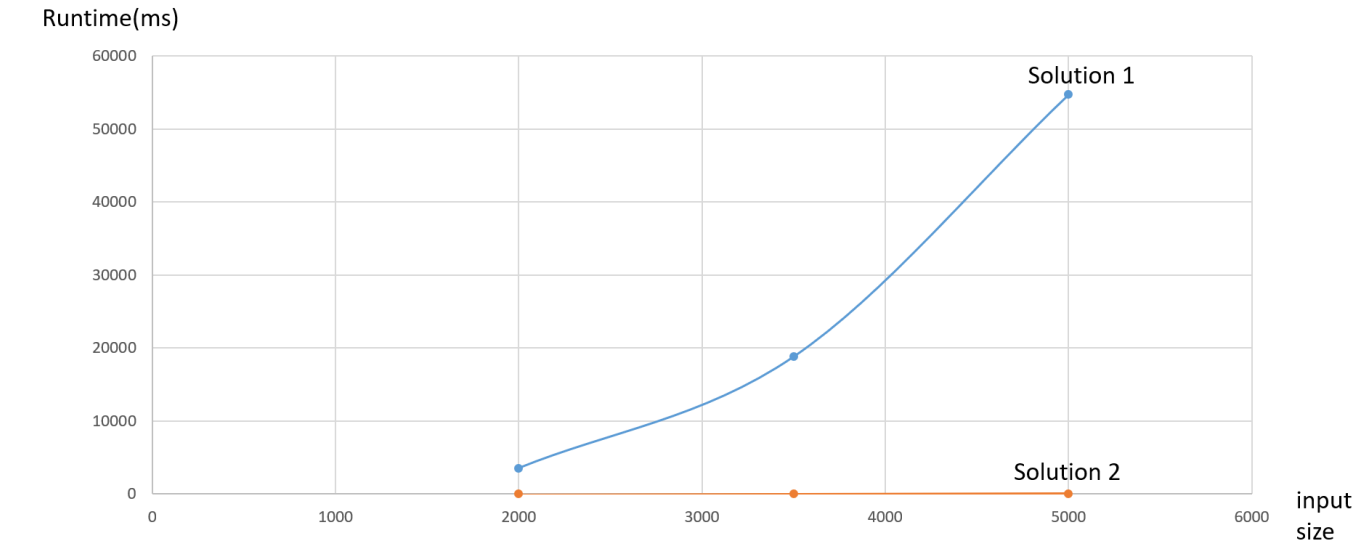
1. Since my computer measures algorithms that their runtimes are less than 1ms as 0ms I could not use same values for all four different solutions. For example, I could measure runtime of 1st solution with a vector that includes 100 items. However, I could not use same vector for solution 4 because it's runtime is too short to be measured. Also, I did use 1million sized vector for solution 4 but I could not use the same vector for solution 1 because it's runtime would be very long that can be measured. So, I prepared one table and plot for solution 1 and 2; one table and plot for solution 3 and 4.

	O(N logN)	**O(N)**
N = 1 M	110 ms	6.3 ms
N = 5 M	577 ms	31.2 ms
N = 10 M	1188.7 ms	62.4 ms

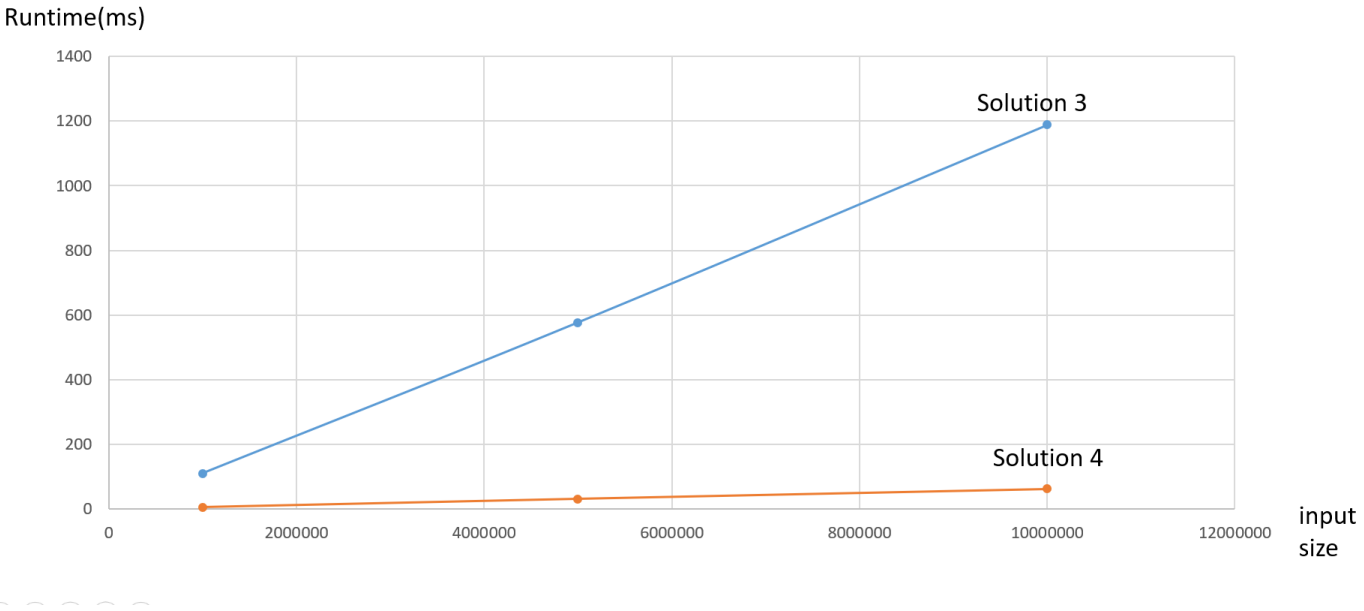
Table 1

	O(N^3)	**O(N^2)**
N = 2000	3537 ms	9.9 ms
N = 3500	18798 ms	28.9 ms
N = 5000	54705 ms	58.8 ms

Table 2


Runtime(ms)
60000
50000
40000
30000
20000
10000
0
0
1000
2000
3000
4000
5000
6000
Solution 1
Solution 2
input size

Plot 1


Runtime(ms)
1400
1200
1000
800
600
400
200
0
0
2000000
4000000
6000000
8000000
10000000
12000000
Solution 3
Solution 4
input size

Plot 2

2. Theoretical growth rates of each solutions:
    - Solution 1 -> O(N^3)
    - Solution 2 -> O(N^2)
    - Solution 3 -> O(N * log N)
    - Solution 4 -> O(N)

When we evaluate the plots and tables, we can see that results are consistent with theoretical analysis. Let us analysis some values:

For solution 1 N is incremented from 2000 to 5000. The ratio is 2/5. The cube of ratio is 8/125. The ratio of runtime is 3537/54705 these ratios are nearly same, so it shows that solution 1 is O(N^3). It's graph is in the shape of x^3 graph.

This time let us evaluate and compare runtimes of different solutions with same value. For this, we will evaluate runtimes of solution 3 and 4 for N = 1.000.000. Runtime of solution 4 is 6.3ms and runtime of solution 3 is 110ms. If we look theoretically, we can think there is a problem. Because solution 3 is O(N * log N) and solution 4 is O(N). Their ratio must be 1/ log N, for N = 1.000.000 this ratio is 1/6. But their ratio is 6/110. It seems like there is a problem. However, there is no problem, everything is alright. Because Big Oh Notation does not include everything. Like some constant multipliers. So, their runtimes' ratio can be different from their Big Oh Notations' ratio. There can be some other constants. Important factor is not consistency with different solutions, important factor is consistency between values of one solution.

Let us evaluate efficiency of solutions. For small values there is no significant difference of their efficiency, even if it is possible to solution 3 is more efficient than solution 4 (for N < 10). However, for larger values there is huge difference between efficiencies. Even computer cannot measure runtime of solution 1 for some N values that solution 4 can easily overcome with it. Like 1M. Solution 4 runs it in just 6.3 ms. For solution 1, I could not see a value in the screen even I waited for long time.

So, we can summarize the efficiency levels like that

S4 > S3 > S2 > S1.

3.

Specifications of my computer:

OS: Windows 10

CPU: Intel Core i7-7700HQ 2.80GHz

RAM: 16 GB

64 Bit

GPU: NVIDIA GeForce GTX 1050

Asus VivoBook Pro