Question-1:

n		10^2	10^3	10^4	10^5	10^6
$2n \log_e n$		921.03	13815.51	184206.81	2302585.09	27631021.12
		μs	ms	ms	ms	s
Average running time of Quick Sort		157.109	2.368	37.889	474.712	5.934
Average running time of Randomized Quick Sort		229.914	2.847	33.165	548.215	8.730
Average no. of comparisons during Quick Sort		401.0	6450.0	75952.0	1041672.0	13279422.0
Average no. of comparisons during Randomized Quick Sort		390.15	6196.64	84531.34	1073807.25	13046259.98
Average value of double sort time by Quick Sort		1213.45	101.13	16686.46	182539.76	1841.07
Average value of double sort time by Randomized Quick Sort		429.32	5.23	64.99	1094.50	17.86
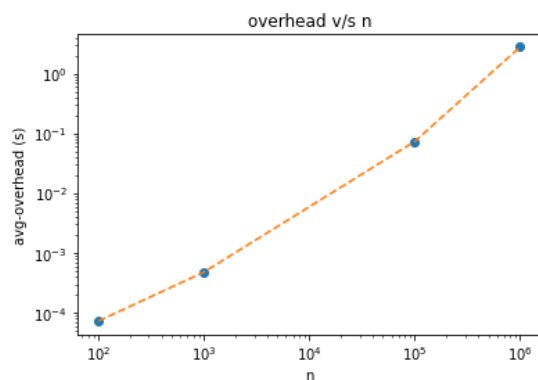
overhead v/s n

Randomized quick-sort in most cases on average takes longer than deterministic quick-sort to sort the array.

Defining the average overhead as the difference between the average runtimes of quick-sort and Randomized Quick-sort. We expected the average-overhead to increase linearly with n.

Since we were able to repeat the experiments only 500 times due to compute and runtime limits, there is a little variation in the recorded runtimes. If we can drop the negative average overhead for $n = 10^4$, we can roughly see a linearly increasing trend.

Qusetion-2:

n		10^2	10^3	10^4	10^5	10^6
		μs	ms	ms	ms	s
Average running time of Quick Sort		157.109	2.368	37.889	474.712	5.934
Average running time of Merge Sort		288.634	3.989	56.841	963.512	13.407
Average no. of comparisons during Quick Sort		401.0	6450.0	75952.0	1041672.0	13279422.0
$2n \log_e n$		921.03	13815.51	184206.81	2302585.09	27631021.12
Average no. of comparisons during Merge Sort		1564.0	23988.0	323248.0	4065667.0	48903806.0
$n \log_2 n$		1328.77	19931.57	265754.25	3321928.09	39863137.14
Number of times Merge Sort outperformed Quick Sort		8	9	95	116	42

On average quick-sort runs faster and requires lesser number of comparisons than merge-sort. The run-time of merge-sort is proportional to $2n \log_e n$ with a proportionality constant of around 460. For quick-sort the run-time is proportional to $n \log_2 n$ with a proportionality constant of 300.

MergeSort runtimes

QuickSort runtimes
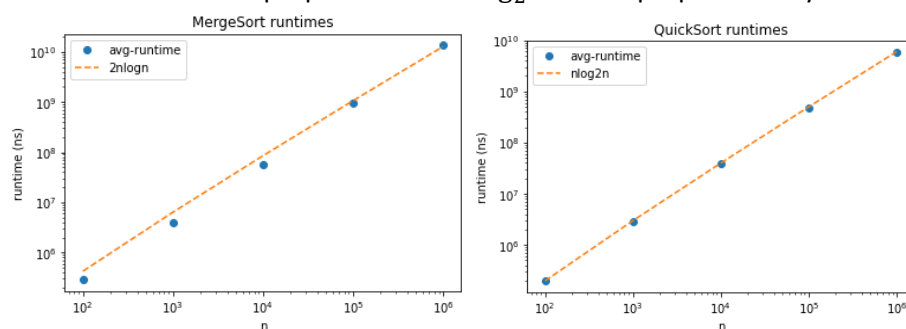
Question-3:

n	10^2	10^3	10^4	10^5	10^6
	μs	ms	ms	ms	s
Average running time of Randomized quick sort	229.914	2.847	33.165	548.215	8.730
$2n \log_e n$	921.03	13815.51	184206.81	2302585.09	27631021.12
Average no. of comparisons during Randomized Quick Sort	390.15	6196.64	84531.34	1073807.25	13046259.98
No. of cases where run time exceeds average by 5%	47	77	96	180	162
No. of cases where run time exceeds average by 10%	21	33	64	129	151
No. of cases where run time exceeds average by 20%	9	11	36	79	129
No. of cases where run time exceeds average by 30%	6	8	24	50	113
No. of cases where run time exceeds average by 50%	5	4	8	22	70
No. of cases where run time exceeds average by 100%	2	1	0	2	2

Randomized quick-sort shows a large variance for higher values of n (more than 10^3). Average number of comparisons grow steeply beyond 10^4. The most of the runtime fall within the 30% range about the average.

Code: Assignment-1(github.com)