Kartik Raj - 14308

Sanjay Ramesh - 14601

Programming Assignment 4

Quick Sort versus Merge Sort

	n= 10^2	n= 10^3	n= 10^4	n= 10^5	n= 10^6
Average running time of Quick Sort(in ms)	0.008	0.12	1.65	20.75	247.84
Average running time of Merge Sort(in ms)	0.009	0.15	2.29	27.04	267.99
Average number of comparisons during Quick Sort	646	10981	155797	2017443	24796387
The value of 1.39n log_2n	923	13852	184699	2308740	27704880
Average number of comparisons during Merge Sort	541	8707	120444	1536146	18670413
The value of n log_2n	664	9965	132877	1660964	19931568
Number of times Merge Sort outperformed Quicksort(out of 2000)	30	10	1	30	23

Observation: Average runtime of Quicksort is lesser than Mergesort though it makes higher number of comparisons on an average. Quicksort outperforms Mergesort in all most all cases.

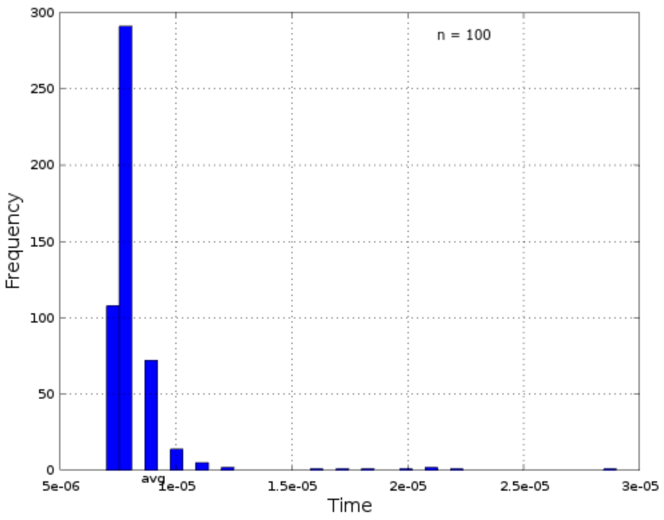
Inference: Reason for this is that the associated time constant(average) with each comparison in mergesort is greater because of too much of data movement operations, resulting in average higher time. Also from the last row, one can infer that most of the cases in quicksort are dense around the "average case" ($O(nlogn)$) and so "worst cases" time ($O(n^2)$) are very less frequent to occur.

Distribution of the running time of Quick Sort (out of 500 cases)

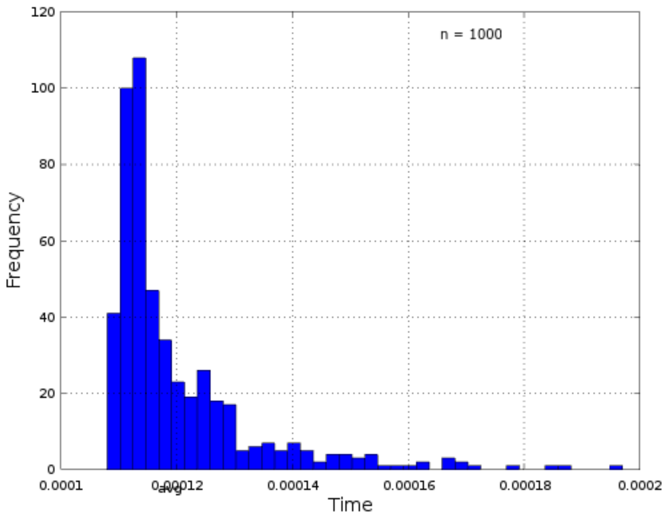
Part (a)	n= 10^2	n= 10^3	n= 10^4	n= 10^5	n= 10^6
Average running time of Quick Sort(in ms)	0.008	0.12	1.65	20.75	247.84
The value of 1.39n log_2n	923	13852	184699	2308740	27704880
Average number of comparisons during Quick Sort	646	10981	155797	2017443	24796387
No. of cases where run time exceeds average by 5%	93	138	195	308	102
No. of cases where run time exceeds average by 10%	21	84	116	122	46
No. of cases where run time exceeds average by 20%	7	41	51	36	10
No. of cases where run time exceeds average by 30%	7	20	8	4	1
No. of cases where run time exceeds average by 50%	7	4	0	0	0
No. of cases where run time exceeds average by 100%	7	0	0	0	0

The observed average number of comparisons in quicksort are slightly lesser than $1.39nlog_2n$. This is because the actual average is $1.39nlog_2n - 2.84n$. This comes out to be close to the observed value.

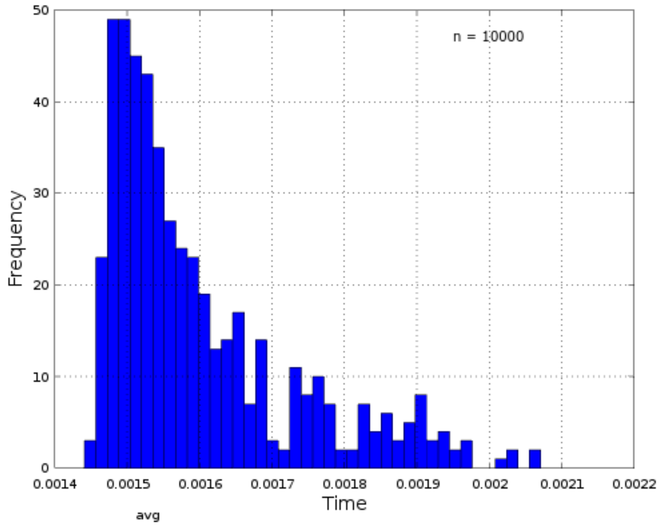
From the table, we see that there are very less cases that exceed the average run time by a significant amount. This further reinforces our claim made from table 1 that there are very few cases that take worst run time.



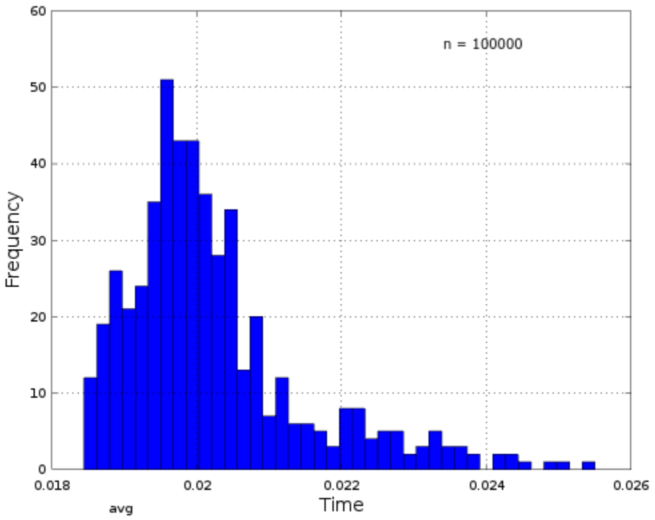
For n = 100



For n = 1000

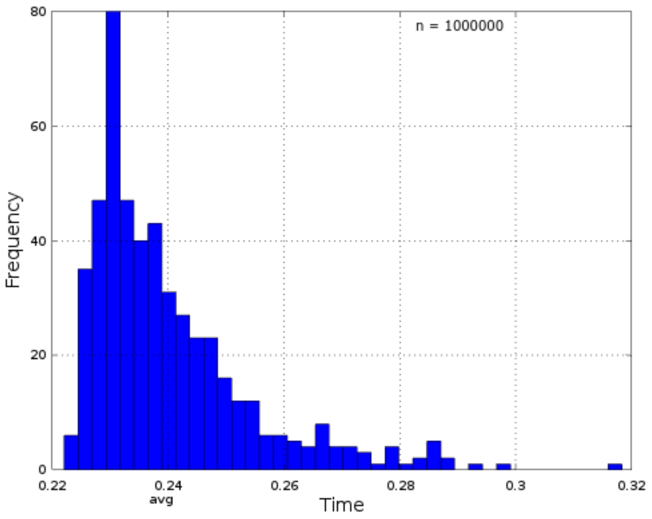


For n = 10000



For n = 100000

The above histograms show that the run time in most cases is centered around average. And we know that average run time of quicksort is better than that of mergesort. So, most of the times in practice the run time of quicksort is better than that of merge-sort, and therefore quicksort is preferred over merge-sort.



For n = 1000000