ESO207: Data Structures and Algorithms

Assignment 5

Soham Bharambe, 210264
Anupam Chaudhary, 210170

November 4, 2022

1. Quick Sort versus Merge Sort

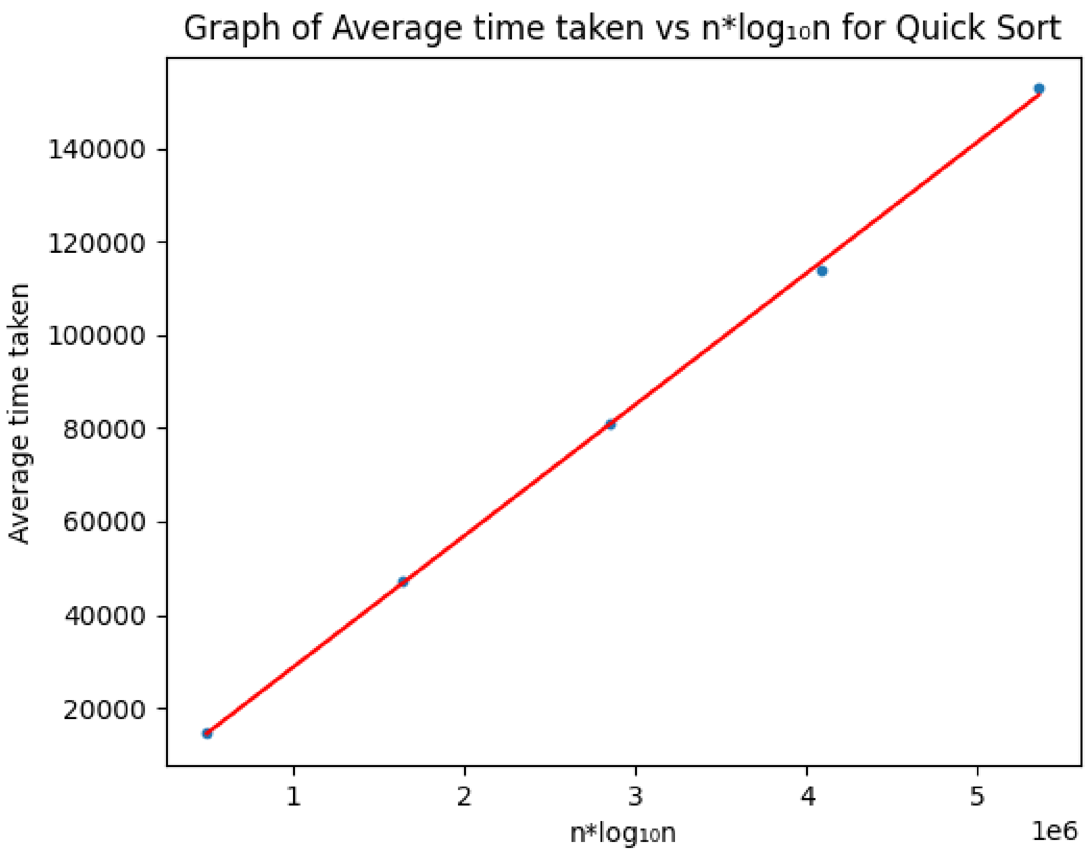
1.1 Comparisons

$n \longrightarrow$	10^2	10^3	10^4	10^5	10^6
Average number of comparisons during Quick Sort	652.686	10,998.044	155,945.346	2,018,511.24	24,763,393.32
$2n \log_e n$	921.034	13,815.511	184,206.807	2,302,585.093	27,631,021.116
Average number of comparisons during Merge Sort	541.608	8,706.744	120,454.832	1,536,398.114	18,674,382.5
$n \log_2 n$	664.386	9,965.784	132,877.124	1,660,964.047	19,931,568.569

By fitting on a graph,
For Quick Sort, average number of comparisons= $2n \log_e n - 3n$
For Merge Sort, average number of comparisons= $n \log_2 n - 1.2n$
This can also be observed from the above table.

1.2 Number of comparisons and time complexity of quick sort

$n \longrightarrow$	1×10^5	3×10^5	5×10^5	7×10^5	9×10^5
Average running time of Quick Sort	14,772.486	47,143.596	80,762.434	113,697.808	152,803.898

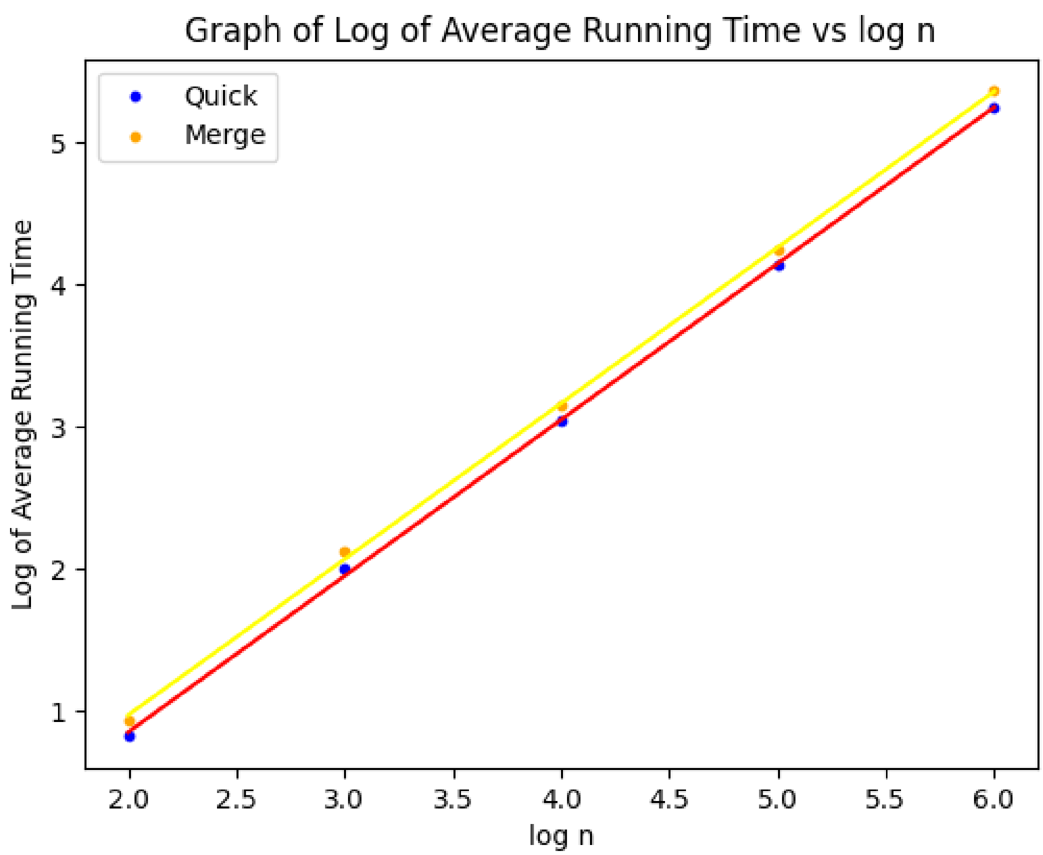


One can see that average running time is of the order $n \log_{10} n$.
This is in line with the theory we studied in class.

*All times are in microseconds

1.3 Time Complexity

$n \longrightarrow$	10^2	10^3	10^4	10^5	10^6
Average running time of Quick Sort	6.692	99.716	1,107.128	13,673.864	177,146.692
Average running time of Merge Sort	8.728	133.778	1,432.652	17,562.664	229,766.326
Number of times Merge Sort outperformed Quick Sort	6	0	0	0	0



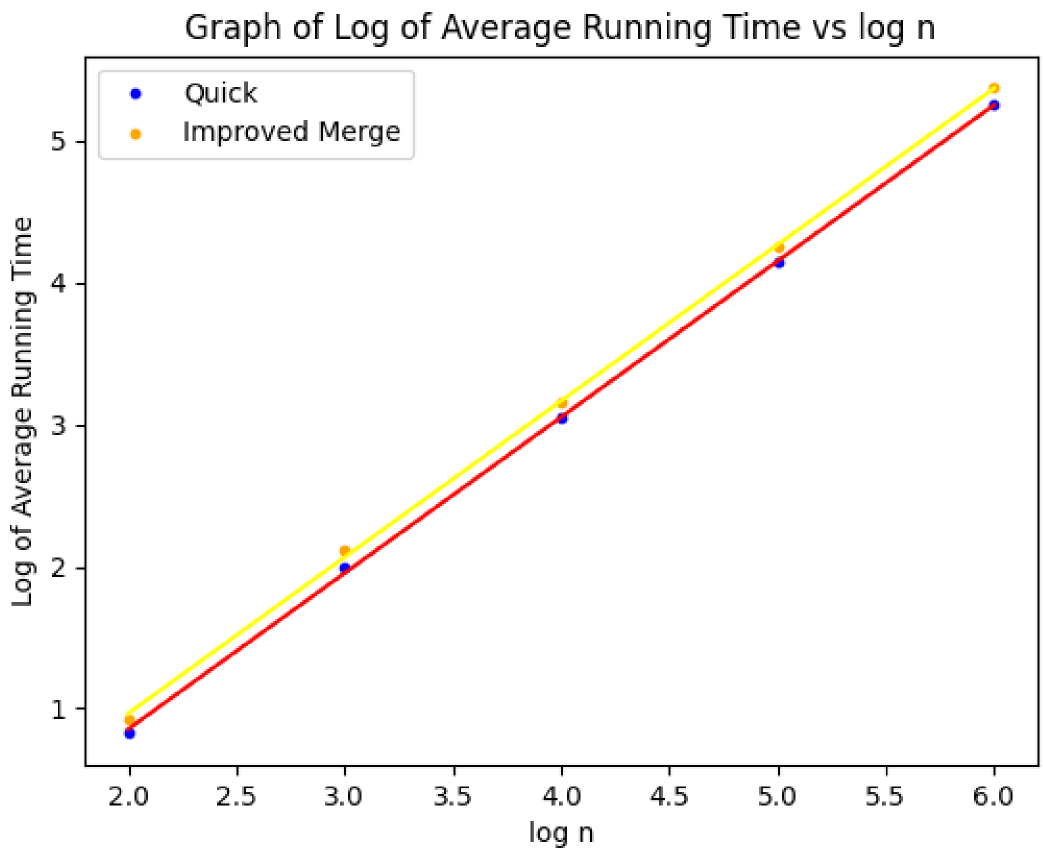
Slope of both lines is the same and only the intercept varies.

This shows that both the algorithms have same order and the only difference of lower order terms, also the coefficient of all terms.

*All times are in microseconds

1.4 Can you improve merge-sort ?

$n \longrightarrow$	10^2	10^3	10^4	10^5	10^6
Average running time of Quick Sort	6.692	99.716	1,107.128	13,673.864	177,146.692
Average running time of Improved-Merge Sort	8.542	131.012	1,434.958	17,738.022	232,644.414
Number of times Improved-Merge Sort outperformed Quick Sort	8	0	0	0	0



Same conclusions as above, but the difference in intercepts this time is less than the one observed above which goes to show that improved merge is faster, bringing both the parallel lines closer.

*All times are in microseconds

2. Reliability of Quick Sort

$n \longrightarrow$	10^2	10^3	10^4	10^5	10^6
Average running time of Quick Sort	6.692	99.716	1,107.128	13,673.864	177,146.692
No. of cases where run time exceeds average by 5%	92	230	24	7	138
No. of cases where run time exceeds average by 10%	92	78	13	5	123
No. of cases where run time exceeds average by 20%	9	14	11	2	15
No. of cases where run time exceeds average by 30%	9	0	0	0	5
No. of cases where run time exceeds average by 50%	6	0	0	0	1
No. of cases where run time exceeds average by 100%	1	0	0	0	0

Only a few cases (about 10%) require more time than average of which almost no array sorting time exceeded 30% of average.

Most of the cases finish faster than the average time which tells us that even if order of merge sort and quick sort average time is same, quick sort still performs better for most cases.

Hence, Quick Sort is relied upon rather than merge sort which has a highly stable average time complexity.

*All times are in microseconds