Assignment2

Subhrajyoty Roy (Roll No. - BS1613)

August 11, 2018

Introduction

This Assignment comprises of the simulation of different sorting techniques like Merge Sort, Quick Sort and Heap Sort. The number of comparisons made and the time taken by these algorithms are obtained for different sizes of inputs.

Simulation Results

- The number of comparison is the average number of comparisons that the particular algorithm took to sort 25 randomly generated arrays of a given size.
- The time is the total amount of time in milliseconds taken by the algorithm to sort 5,000 randomly generated arrays of a given size.

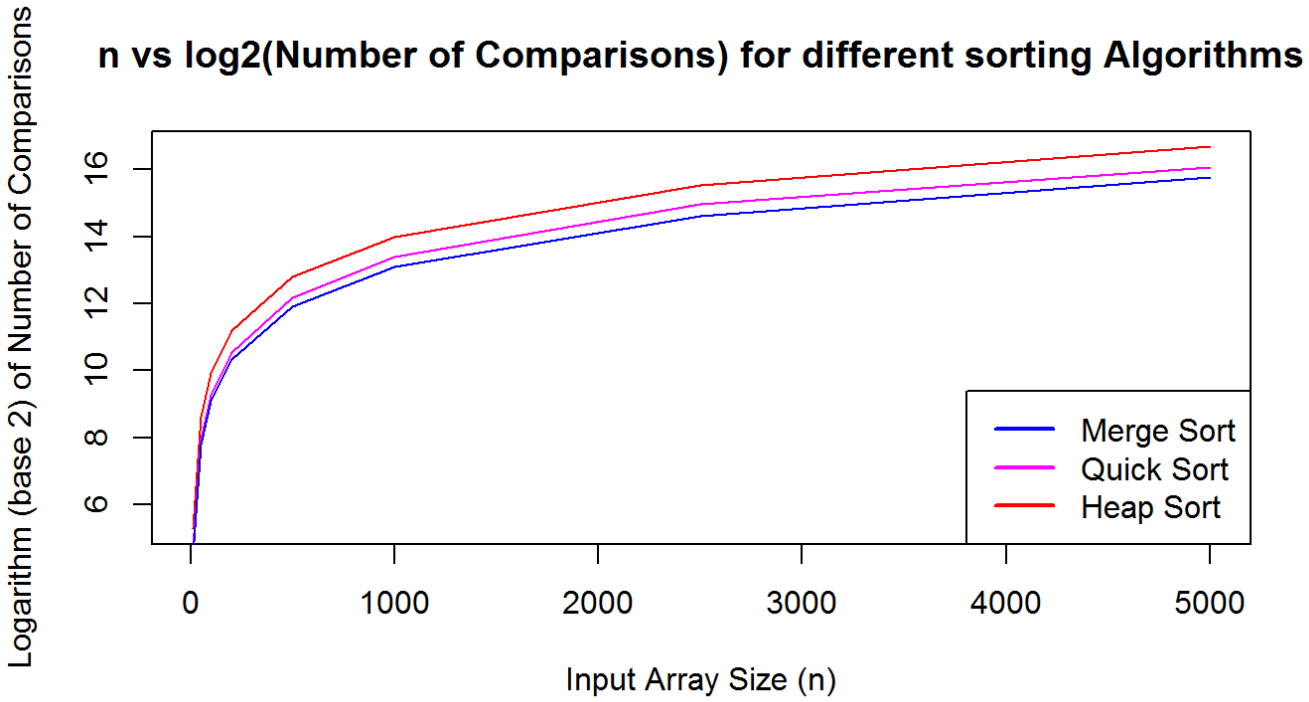
N	Mergesort.Comparison	Quicksort.Comparison	Heapsort.Comparison
10	23	27	39
50	221	252	389
100	541	616	970
200	1282	1506	2338
500	3854	4592	7122
1000	8707	10670	16251
2500	25113	31737	47360
5000	55226	68915	104687

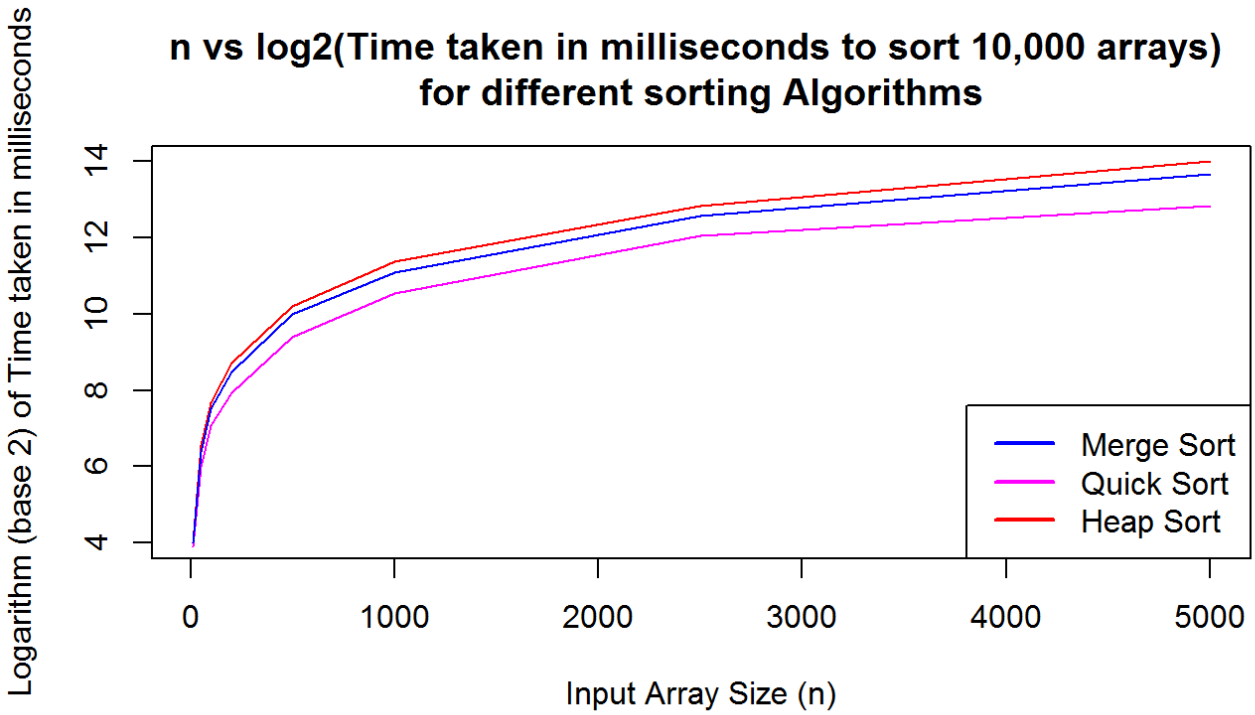
N	Mergesort.Time	Quicksort.Time	Heapsort.Time
10	16	15	16
50	83	63	94
100	185	134	203
200	360	243	422
500	1023	678	1186

N	Mergesort.Time	Quicksort.Time	Heapsort.Time
1000	2162	1494	2632
2500	6124	4203	7305
5000	12876	7305	16293

Figures

The plots indicates the graph of size of the array (n) vs the logarithm of the complexity of different sorting algorithms, namely with respect to number of Comparisons made and with respect to time.





Conclusions

- Theroetically, Heap sort is asymptotically better than Merge sort and Quick sort. However, the above results show that Heap Sort uses more comparisons than Quick sort and Merge Sort. This is because Heap sort usually performs better for large values, possibly much larger than $n = 5000$.
- Although Merge Sort uses less number of comparisons than Quick sort, quick sort is indeed quick (as uses less amount of time). This is because the merging step in the Merge sort is not done in place, hence the time taken for copying the merged array to the actual array is also included in the mergesort time.

THANK YOU