Name : Mayuresh Bhagwan Nehe

PRN No: - 20190802076

LAB-2

Aim : Implementation of two sorting algorithm, quick sort and merge sort using iterative and recursion method. Additionally, we would be also computing the memory utilization for each program

1.Quick Sort :

Quicksort is a divide-and-conquer algorithm. It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays.

Complexity of Quick sort :

Worst complexity: n^2

Avg complexity: n*log(n)

Best complexity: n*log(n)

2.Merge Sort :

Merge sort is an efficient, general-purpose, comparison-based sorting algorithm. It works by Dividing the unsorted list into *n* sublists, each containing one element and repeatedly merge sublists to produce new sorted sublists until there is only one sublist remaining.

Complexity of Merge sort :

Worst complexity: n*log(n)

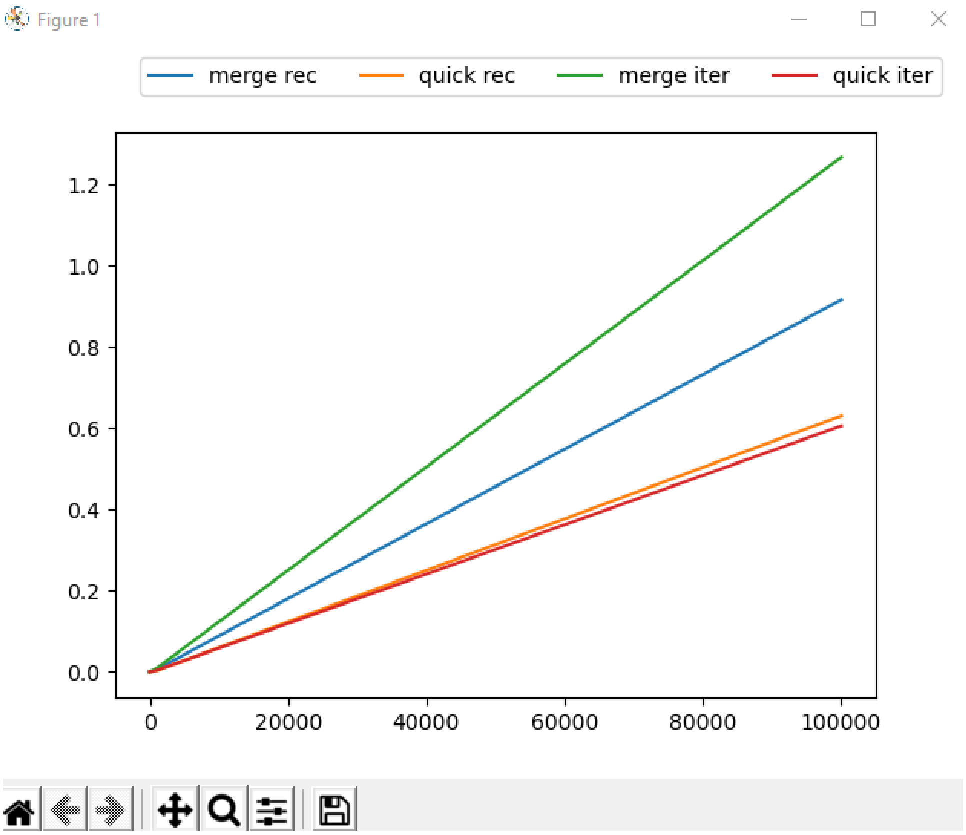
Avg complexity: n*log(n)

Best complexity: n*log(n)

The Table below gives time(s) taken by each sort for values of n :

Sort (n)	10	100	1000	10000	1000000
Quick (rec)	0.0	0.0	0.0039	0.037	7.59
Quick al(iter)	0.0	0.0	0.0039	0.041	8.10
Merge (rec)	0.0	0.0	0.0059	0.071	13.90
Merge (iter)	0.0	0.0	0.0089	0.106	18.45

The Graph for the above table is given below:



Computing the Memory taken for n=100000 :

	Total Memory used (MB)	Actual Memory used (MB)
Initial Before Sort	58.8	0
Quick Sort (rec)	58.7	0.01
Merge Sort (rec)	58.8	0
Quick Sort (iter)	58.7	0.01
Merge Sort (iter)	58.8	0