ESO207A
Programming Assignment 3

Deepanshu Bansal(150219)
Mukul Chaturvedi(150430)

Q-1

Number of different random arrays = 1000.

	n=10^2	**10^3**	**10^4**	**10^5**	**10^6**
Average running time of QuickSort	0.000013	0.000165	0.002213	0.027766	0.189378
Average running time of MergeSort	0.000019	0.000219	0.002824	0.034397	0.222822
Average number of comparisons in QuickSort	645	10978	155518	2016884	189733205
Average number of comparisons in MergeSort	542	8707	120453	1536397	186744521
No. of times MergeSort had lesser no. of comparisons than QuickSort	1000	1000	1000	1000	970

Explanation

Average Time complexity of both algorithms is **O(nlogn)** while worst case time complexity of Quicksort is **O(n^2)** with **O(1)** extra space and mergesort takes **O(N)** extra space . Now number of comparisons in Quicksort is more than Mergesort despite of this quicksort has lesser running time because mergesort takes **O(N)** extra space and allocating and de-allocating the extra space used increases the running time of the algorithm .

Number of different random arrays = 1000.

	n=10^2	**10^3**	**10^4**	**10^5**	**10^6**
Average running time of QuickSort	0.000013	0.000165	0.002213	0.027766	0.189378
Average number of comparisons in QuickSort	645	10978	155518	2016884	189733205
Percentage of cases when running time of QuickSort exceeds average by 5%	12.1	3.6	7.1	10.6	12.2
Percentage of cases when running time of QuickSort exceeds average by 10%	3.1	0.8	3.1	6.5	9.4
Percentage of cases when running time of QuickSort exceeds average by 20%	1.4	0.2	1.6	1.5	0.0
Percentage of cases when running time of QuickSort exceeds average by 30%	0.6	0.0	0.5	0.2	0.0
Percentage of cases when running time of QuickSort exceeds average by 40%	0.3	0.0	0.2	0.0	0.0
Percentage of cases when running time of QuickSort exceeds average by 100%	0.2	0.0	0.1	0.0	0.0

Explanation

Average Time increases as average time complexity of algorithm is **O(nlogn)** and as n increases number of comparisons increases and hence the time increases. For a particular n percentage of cases when running time of QuickSort exceeds average by x% decreases as value of x increases because for Quicksort we have an random array for every case and for this algorithm average case time is more or less same for an random input and thus spread of time for various random inputs is concentrated around some area thereby decreasing the number of cases with very huge difference from the average running time and hence as time difference increases from from average the time complexity decreases.

Q-2

Time complexity analysis of **Op function**

$$T(n) = 3T(n/3) + O(n)$$

$$= 3T(n/3) + n$$

$$= 3(3T(n/3^2) + n/3) + n$$

$$= 3^2T(n/3^2) + 2n$$

$$\ldots\ldots$$

$$= 3^kT(n/3^k) + kn$$

Let $n = 3^k$ and $T(1) = c$ and hence $k = \log_3 n$

$$T(n) = nT(1) + n\log_3 n$$

$$T(n) = O(n\log n)$$

Hence time complexity of **Op** function is **O(nlogn) .**

Now for the main function for **q queries** the total time is

$$T(n) = T(op) + qT(query)$$

And T(query) in worst case is **O(1)** as we have to to just calculate (**sum[j]-sum[i-1])** as l and r are fixed as we have done pre-processing.

So,

$$T(n) = T(op) + qT(query)$$

$$= O(n\log n) + qO(1)$$

$$\mathbf{T(n) = O(n\log n + q)}$$

-----THE-END-----

Deepanshu Bansal(150219)