REPORT

on

Assignment 7

Name: Abdus Samee

ID: 1805021

Course: CSE - 204

Topic: Sorting Algorithms

COMPLEXITY ANALYSIS:

Mergesort:

Mergesort is a sorting algorithm which executes the divide and conquer approach. The divide step occurs in order of $\Theta(1)$ whereas the conquer step occurs in the order $\Theta(n)$. So, the recurrence relation becomes as follows:

$$T(n) = \Theta(1) \text{ , when } n = 1$$
$$T(n) = 2T(n/2) + \Theta(n) \text{ , when } n > 1$$

Using master theorem, from this recurrence relation, we get a = 2, b = 2, f(n) = n. Thus, $n^{\log_b a} = \Theta(n)$. Since $f(n) = \Theta(n)$, from condition 2, we get the order of merge sort as $\Theta(n\log n)$, which can also be termed in this case as $O(n\log n)$. The worst case running time as well as the best case running time for the mergesort is always the same. So for the arrays of ascending, descending and random order, mergesort takes similar amount of time.

Mergesort requires additional array while merging the sorted sub-arrays. So, the space complexity of merge sort is $O(n)$ if there is no memory leak. If there is some memory leak, it might turn into $O(n\log n)$.

Quicksort:

Quicksort is also a sorting algorithm which follows the divide and conquer rule. Here, the divide step or partitioning take more time than conquer step compared to mergesort. But its running time differ depending on the order of the array (ascending, descending or random distribution of elements).

In worst case, the array is already sorted in ascending or descending order. In this case, unbalanced partitioning occurs in divide step. The recurrence relation becomes:

$$T(n) = T(n-1) + \Theta(n)$$

which results in $\Theta(n^2)$.

In best case, the partitioning produces two sub-problems each of size not more than n/2. The recurrence relation then becomes:

$$T(n) = 2T(n/2) + \Theta(n)$$

From the second condition of master theorem, it becomes $\Theta(n\log n)$.

Quicksort uses in-place sorting. Each recursive call of quicksort uses a constant space of O(1). In average case, the space complexity becomes O(log n), but in worst case it becomes O(n).

MACHINE CONFIGURATION:

OS: Windows 10 (64 bit)
Model: ASUS Vivobook S15 X520UN
Processor: Intel® Core™ i5-8250U CPU@ 1.60GHz(8 CPUs)
Memory: 8192 MB RAM

TABLE:

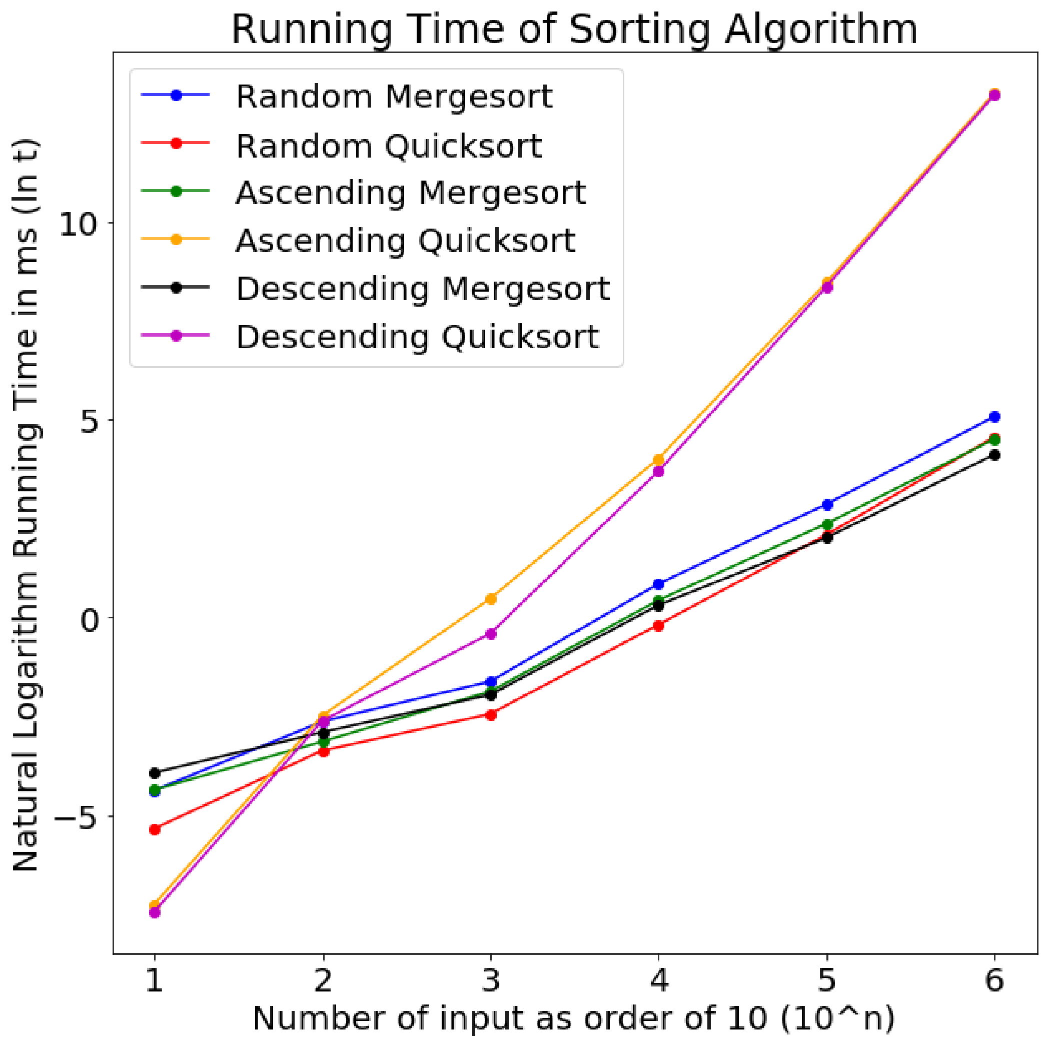
The average time to accomplish both the mergesort and quicksort algorithm is shown in the table as follows:

Table: Average time for sorting n integers in different input orders (ms)

Input Order	n= Sorting Algorithm	10	100	1000	10000	100000	1000000
Ascending	Merge	0.0127	0.043	0.1526	1.5269	10.6043	90.3085
	Quick	0.0007	0.0825	1.5917	54.7367	4706.3944	569942.8376
Descending	Merge	0.01955	0.0545	0.1404	1.3501	7.4088	61.067
	Quick	0.00058	0.0722	0.6582	39.8678	4253.3446	538188.8055
Random	Merge	0.0125	0.0713	0.1966	2.3127	17.3816	158.6716
	Quick	0.00475	0.0342	0.0863	0.8247	8.0049	94.9882

PLOT:

The plot containing the average running time of both the sorting algorithms is as follows:



Running Time of Sorting Algorithm
Random Mergesort
Random Quicksort
Ascending Mergesort
Ascending Quicksort
Descending Mergesort
Descending Quicksort
Natural Logarithm Running Time in ms (ln t)
10
5
0
−5
1
2
3
4
5
6
Number of input as order of 10 (10^n)