CS2023 - Data Structures and Algorithms
In-class Lab Exercise – Lab Report

Week 5

By: Sajeev Kugarajah (210554M)

Github directory link:
https://github.com/veejask-41/210554M-CS-2023-Data_Structures_And_Algorithms/tree/main/week%2005/lab%2005

Question 01)

SOURCE CODE IS UPLOADED TO THE GITHUB REPOSITORY

Time taken for recursive quick sort algorithm in nanosecconds

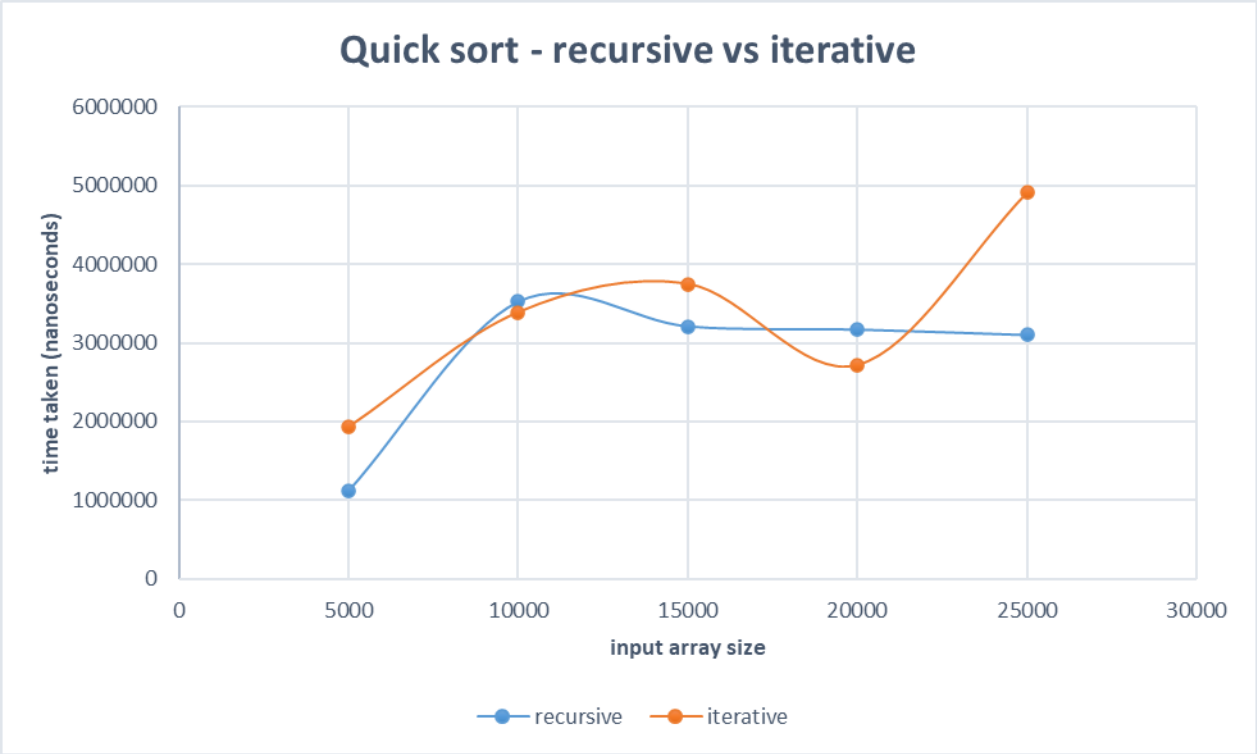
arr size	test 1	test 2	test 3	test 4	test 5	average
5000	1040000	1506000	515000	2010000	510000	1116200
10000	1017000	8012000	1508000	3568000	3519000	3524800
15000	5073000	3513000	1700000	2764000	3010000	3212000
20000	3530000	6089000	4012000	512000	1709000	3170400
25000	6052000	2009000	506000	3535000	3400000	3100400

Time taken for iterative quick sort algorithm in nanoseconds

arr size	test 1	test 2	test 3	test 4	test 5	average
5000	505000	2130000	513000	2512000	4017000	1935400
10000	2512000	1024000	3025000	4011000	6384000	3391200
15000	2028000	4011000	5188000	4012000	3512000	3750200
20000	508000	2028000	4980000	3041000	3055000	2722400
25000	3677000	3846000	5520000	8098000	3409000	4910000

Timing plot using average times

arr size	recursive	iterative
5000	1116200	1935400
10000	3524800	3391200
15000	3212000	3750200
20000	3170400	2722400
25000	3100400	4910000



Discussion

This graph illustrates that both of the recursive and iterative algorithms are taking approximately equal time except in some cases. And time taken for both algorithms are increasing with increase of array size.

In some cases iterative algorithm is taking a longer time to sort the array, the array which was inputted was already partially sorted can be a valid reason for that because recursive algorithm works more efficient for partially sorted arrays than iterative algorithm as it is using divide and conquer method, partitioning the whole array.And for largest input sizes with worst cases, the iterative algorithm will be faster as it is avoiding the stack overhead unlike recursive algorithm.

Both of the quick sort algorithms have same worst and average case time complexity of o(n.logn) and $o(n^2)$.

Question 02)

SOURCE CODE IS UPLOADED TO THE GITHUB REPOSITORY

Command line output