CS2023 - Data Structures and Algorithms
In-class Lab Exercise

Week 5

Index Number : 200087A
Name : Chandrasiri Y.U.K.K.

GitHub Link for codes : https://github.com/kavindukalinga/In20-S4-CS2023.git

(Question 1)

Expected output:

```
PS D:\Academic\ENTC\In20-S4\CS2023 Data Structures and Algorithms\inclass2\code\output> & .\'cpptest.exe'
Total elements in quickSortRecur array: 500  Time taken: 318000 nanoseconds
Total elements in quickSortRecur array: 1000  Time taken: 1246000 nanoseconds
Total elements in quickSortRecur array: 5000  Time taken: 28638000 nanoseconds
Total elements in mquickSortRecur array: 10000  Time taken: 60154000 nanoseconds
Total elements in quickSortRecur array: 20000  Time taken: 125681000 nanoseconds


Total elements in quickSortItter array: 500  Time taken: 108000 nanoseconds
Total elements in quickSortItter array: 1000  Time taken: 1310000 nanoseconds
Total elements in quickSortItter array: 5000  Time taken: 27367000 nanoseconds
Total elements in quickSortItter array: 10000  Time taken: 107218000 nanoseconds
Total elements in quickSortIttersion array: 20000  Time taken: 423098000 nanoseconds
PS D:\Academic\ENTC\In20-S4\CS2023 Data Structures and Algorithms\inclass2\code\output>
```

Recursive-Quick Sort:

Total elements in array:	Time taken in nanoseconds
500	318000
1000	1246000
5000	28638000
10000	60154000
20000	125681000

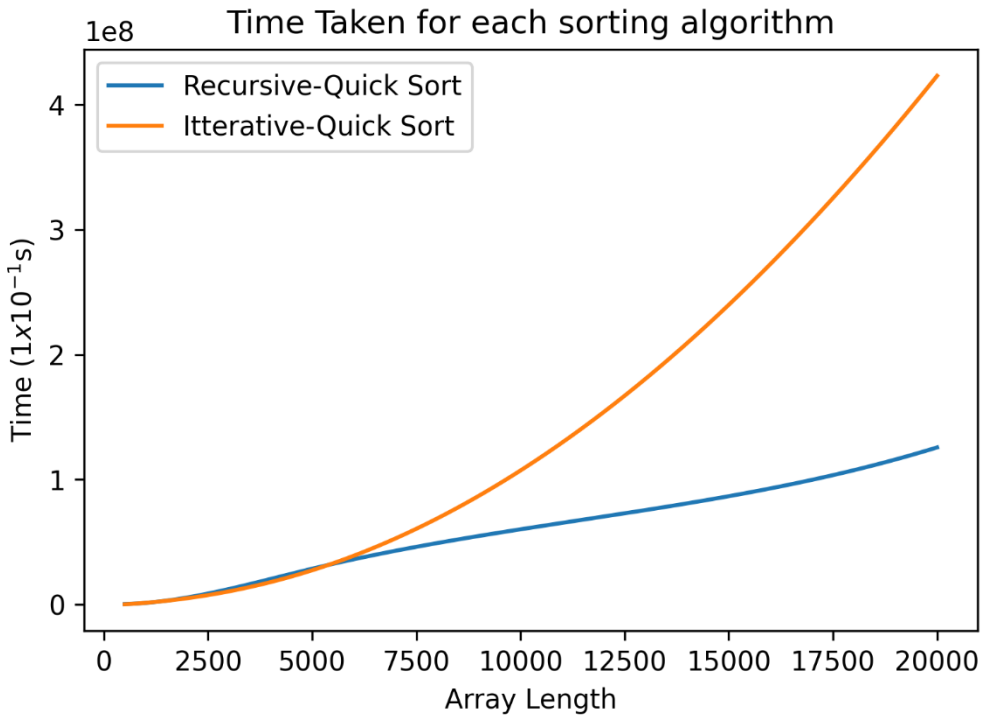
Iterative-Quick Sort:

Total elements in array:	Time taken in nanoseconds
500	108000
1000	1310000
5000	27367000
10000	107218000
20000	423098000

Comparison:

Total elements in array:	Time taken in nanoseconds	
	Recursive	Iterative
500	318000	108000
1000	1246000	1310000
5000	28638000	27367000
10000	60154000	107218000
20000	125681000	423098000

Expected plot:



Comment:

- We can see that for smaller values of Array Length, Iterative Quick Sort is slightly better than Recursive Quick Sort but for larger values of Array Length, Recursive Quick Sort is better in terms of time complexity for these input arrays.