CO322 – Lab 02

Task 3: Finding the performance of the three algorithm.

- Theoretical:

1. Quick Sort:
   Best case = O (n log n) for simple partition and O (N) for three-way partition.
   Worst case = O (N^2)
   Average case = O (n log n)

2. Merge Sort:
   Best case = O (n log n)
   Worst case = O (N)
   Average case = O (n log n)

- Experimental:

According to the theoretical approach, considered algorithms' average time complexity is same.

Case 1:

1. Quick Sort:
   Length 100 of unsorted array,
   Execution time: 0.878230094909668
2. Merge Sort:
   Length 100 of unsorted array
   Execution time: 1.969526767730713

According to this, Quick sort is better than the Merge sort algorithm

Case 2: Unsorted big array

1. Quick Sort:
   Length 1000 of unsorted array,
   Execution time: 0.6833288669586182
2. Merge Sort:
   Length 1000 of unsorted array
   Execution time: 0.8004281520843506

According to this, Quick sort is better than Merge sort. But in quick sort implementation, choosing the pivot would cause this answers.

Case 3: Sorted big array in ascending order

1. Quick Sort:
   Length 1000 of sorted array,
   Execution time: 0.7199976444244385
2. Merge Sort:
   Length 1000 of sorted array
   Execution time: 0.8551497459411621

When considering sorted array with 1000 element, Quick sort got less execution time.

Case 4: Sorted big array in descending order

1. Quick Sort:
   Length 1000 of sorted array (descending order)
   Execution time: 0.6691203117370605
2. Merge Sort:
   Length 1000 of sorted array (descending order)
   Execution time: 1.0320420265197754

When considering case 3 and case 4, Quick sort perform best execution time for best and the worst cases.

When considering theoretical approach, average time complexity of those two algorithms should be same. But in experimental approach it may be same for particular occasions.