REPORT

Student	MavID	Responsibility
Shashwat Shekhar	1001955798	Merge sort and Quicksort, generating input files
Nikhil Ramesh Yadav	1001949220	INSERTION SORT and Reading from input files and writing output to output files.

Time Complexity of Algorithm

1. Insertion Sort

Time Complexity – Best Case	O(n)
Time Complexity – Average Case	$O(n^2)$
Time Complexity – Worst Case	$O(n^2)$

Insertion Sort has best case time complexity when the number of elements in the array is already sorted in the correct order. It is efficient algorithm for sorting small numbers of elements. Whereas in worst case time complexity, the order is n^2 if the number of elements is sorted in descending order. Therefore, all the elements must be shifted from right to left so it's in sorted order. Insertion sort follows incremental approach.

2. Merge Sort

Time Complexity – Best Case	O(nlog(n))
Time Complexity – Average Case	O(nlog(n))
Time Complexity – Worst Case	O(nlog(n))

It is based on Divide and conquer algorithm. In Merge sort, the time complexity of all the case is same because it divides the array into two halves then solve and sort the smaller halves array. Then it merges the half array into one sorted array.

3. Quicksort

Time Complexity – Best Case	O(nlog(n))
Time Complexity – Average Case	O(nlog(n))
Time Complexity – Worst Case	$O(n^2)$

In Quicksort, the array is divided into subarrays by selecting the pivot elements. When dividing, the elements less than pivot go to left and elements greater than pivot are positioned to right. Both left and right are divided using same approach. When the elements are sorted it is combined and formed a sorted array.

Experimental Result:

Algorithm	Array 20	Array 100	Array 1000	Array 4000
Insertion Sort	18900	815200	10586600	47353300
Merge Sort	52400	270800	3583400	6519400
Quick Sort	23400	83500	1610500	1660700

Time in nanoseconds.

Analysis of Experimental Result vs Theoretical Result

As seen in the above table, Insertion sort is faster than merge sort and quick sort for fewer elements (in our case Array of size 20) because insertion sort has better running time with smaller datasets. It is faster than Quicksort, as Quicksort requires an extra overhead from the recursive function calls. It is faster than merge sort as mergesort requires an extra space for copying the array. This is possible only for smaller numbers of elements as for large numbers of elements this constant time-consuming functions are much smaller than $O(n^2)$ average case time complexity.

For datasets with Array 100,1000 and 4000, quicksort is faster than the other two sorting algorithms because merge sort requires additional space as it creates extra array for storing the data. Since we use 2-dimensional arrays for our computation, creating two extra 2-dimensional arrays and copying them takes time, which is why merge sort is significantly slower than quicksort.

REFERENCES:

https://stackoverflow.com/questions/70402/why-is-quicksort-better-than-mergesort

HONOR CODE:

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.
I promise that I will submit only work that I personally create or that I contribute to group collaborations and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.
I will not participate in any form of cheating / sharing the questions / solutions.

Shashwat Shekhar
7th October 2022.

HONOR CODE

I pledge, on my honor, to ~~uphold~~ uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

I will not participate in any form of cheating / sharing the questions / solutions.

DATE: 10/7/22

SIGNATURE: Yadav NIKHIL RAMESH YADAV.