Tamir Michaely

CSS 342 Program 4

Table 1: Data Used for Different Sorting Methods

Sorting Methods	**Size: 100**	**Size: 1,000**	**Size: 10,000**	**Size: 25,000**
Bubble Sort	214	11,096	701,806	4,455,657
Insertion Sort	30	3,977	164,898	1,047,245
Merge Sort	103	1,643	5,920	11,074
Iterative Merge Sort	40	589	4,213	11,610
Quick Sort	26	549	2,955	4,567
Shell Sort	48	700	3,369	9,701

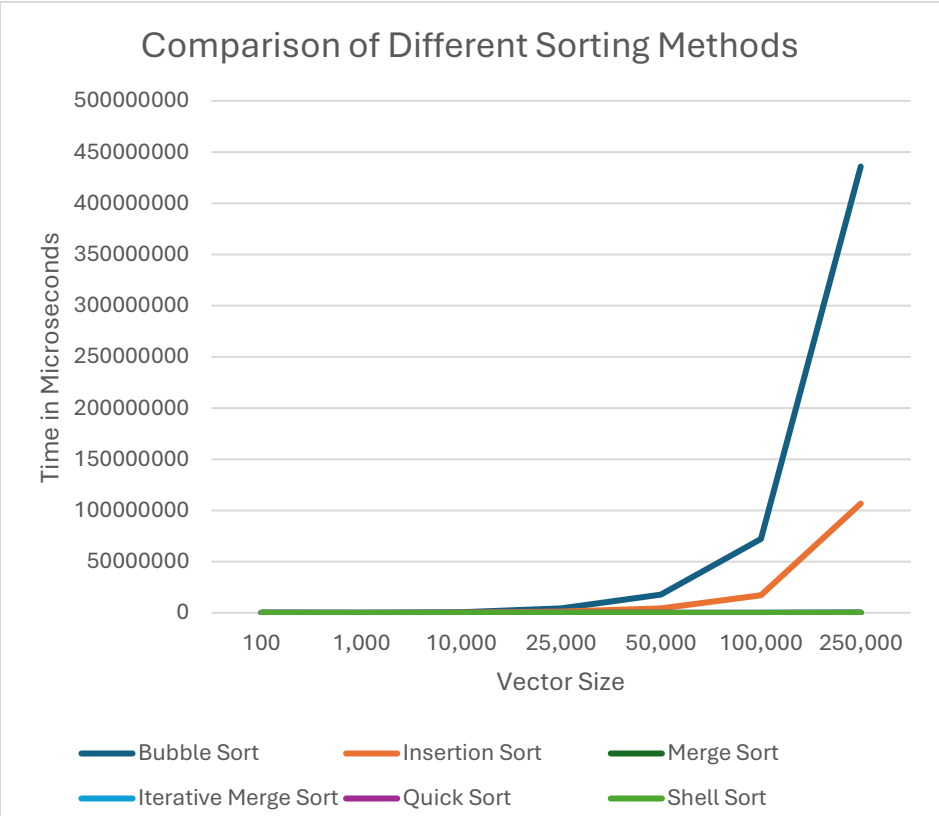
Table 1: Continued

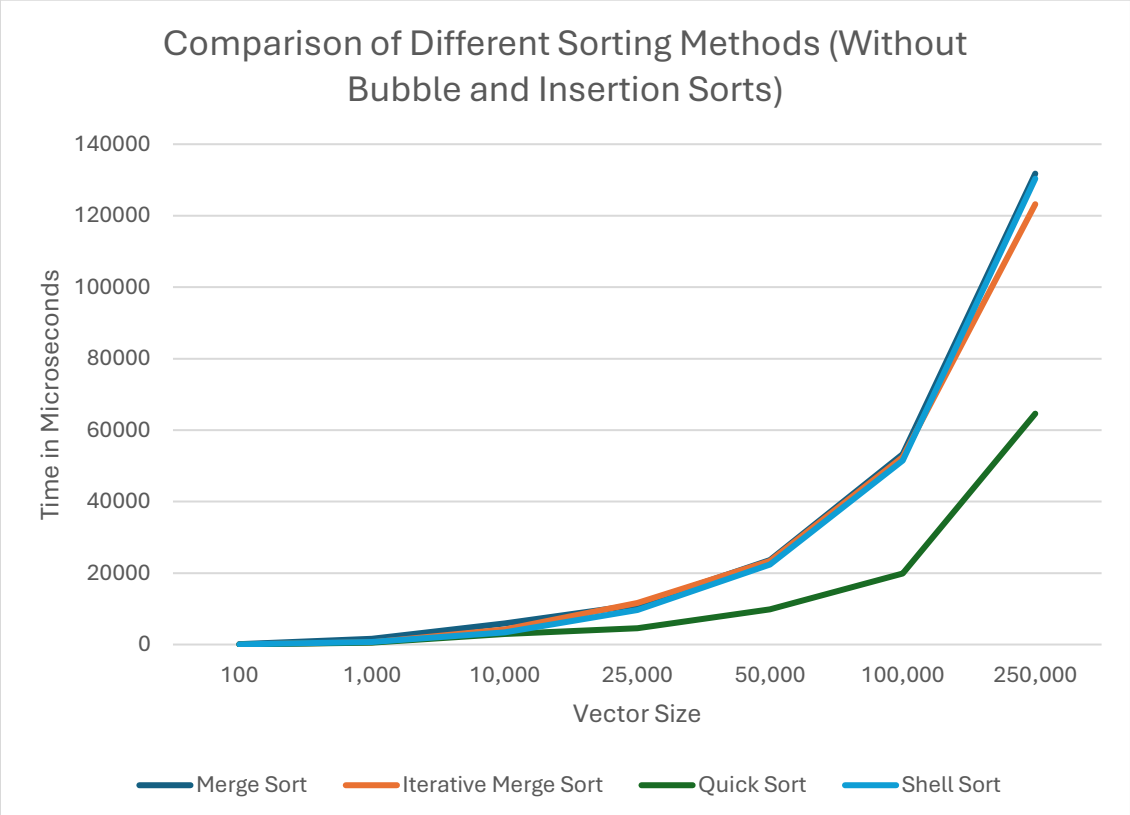
Sorting Methods	**Size: 50,000**	**Size: 100,000**	**Size: 250,000**
Bubble Sort	17,872,320	72,017,908	435,814,816
Insertion Sort	4,190,569	17,036,559	106,678,132
Merge Sort	23,650	53,286	131,769
Iterative Merge Sort	23,393	52,473	123,215
Quick Sort	9,864	19,856	64,608
Shell Sort	22,375	51,442	130,362

The size shows the number of integers in the vector

The time in the tables is measured in microseconds

Graphs Using Table 1 Data:





Conclusion of Graphs:

Looking at the first graph, Bubble Sort and Insertion sort are seen the most due to their average time complexity of O(n^2), so they tend to be the slowest sorting algorithms.

In the second graph, the faster algorithms – Merge Sort, Iterative Merge Sort, Quick Sort, and Shell Sort – stand out due to their more efficient time complexities, such as O(n log(n)).

Quick Sort was found to be the fastest sorting algorithm for every vector size. Iterative Merge Sort was found to be slightly faster than the normal Merge Sort which uses recursion. One reason the Iterative Merge Sort wasn't much faster than the normal Merge Sort could be because it also uses a helper function called Merge to help sort the vector.

Table 2: Big O Complexity of Different Sorts

Sorting Methods	**Best**	**Average**	**Worst**
Bubble Sort	O(n)	O(n^2)	O(n^2)
Insertion Sort	O(n)	O(n^2)	O(n^2)
Merge Sort	O(n log(n))	O(n log(n))	O(n log(n))
Iterative Merge Sort	O(n log(n))	O(n log(n))	O(n log(n))
Quick Sort	O(n log(n))	O(n log(n))	O(n^2)
Shell Sort	O(n log(n))	O($n^{3/2}$)	O(n^2)