Razvan Anghel

12/7/2024

CSS 342

Program 4 Report

Context:

In this lab, we benchmarked 6 different sorting algorithms to illustrate the time complexities of their performance as N approaches infinity. For this lab, I gathered data with N items being the independent variable and time being the independent variable. The data gathered goes up to an N of 75,000 random numbers put into a C++ vector, then sorted by the different algorithms. This was done with a total of 18 data points per sorting algorithm with the average of 10 trials taken per data point.

Theoretical Time Complexities:

Bubble Sort

- Average O(n^2)
- Best Case O(n) (already sorted)
- Worst Case O(n^2)

Insertion Sort

- Average O(n^2)
- Best Case O(n) (already sorted)
- Worst case O(n^2)

Merge Sort

- Every Case O(n*logn)

Iterative Merge Sort (uses less auxiliary space)

- Every Case O(n*logn)

Quick Sort

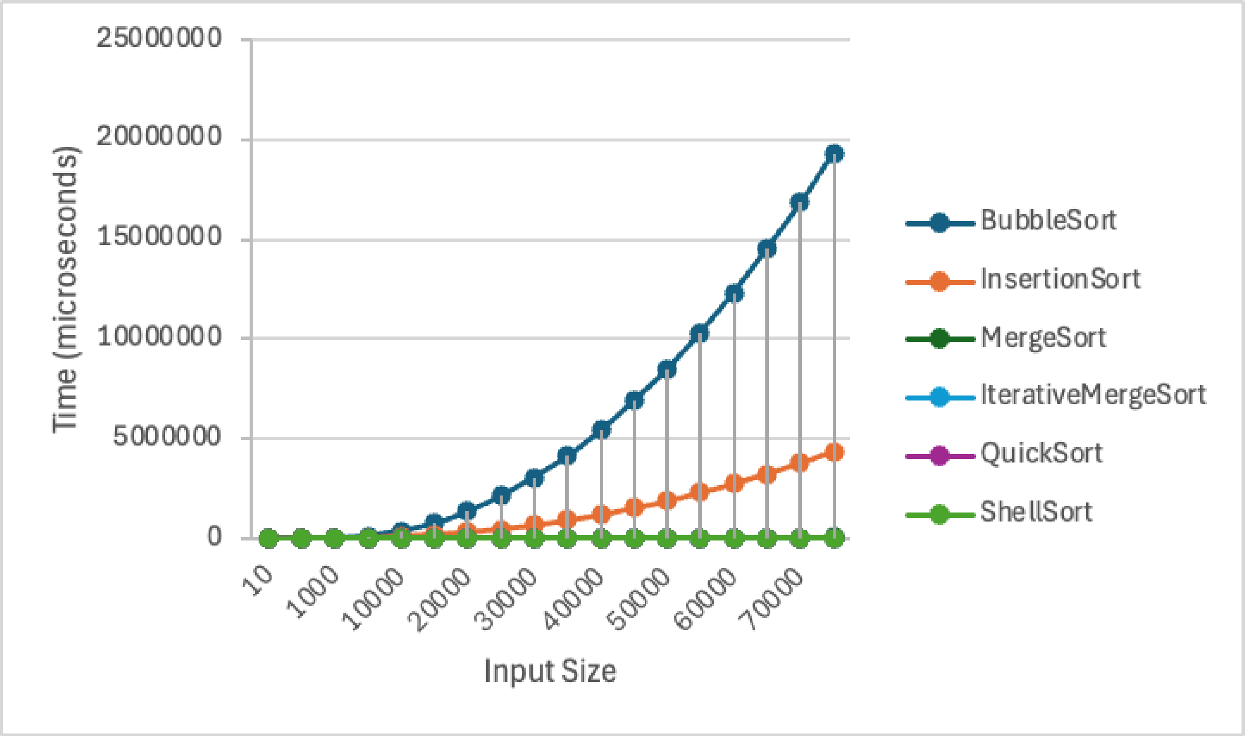
- Average & Best Cases O(n*logn)
- Worst Case O(n^2)

Shell Sort

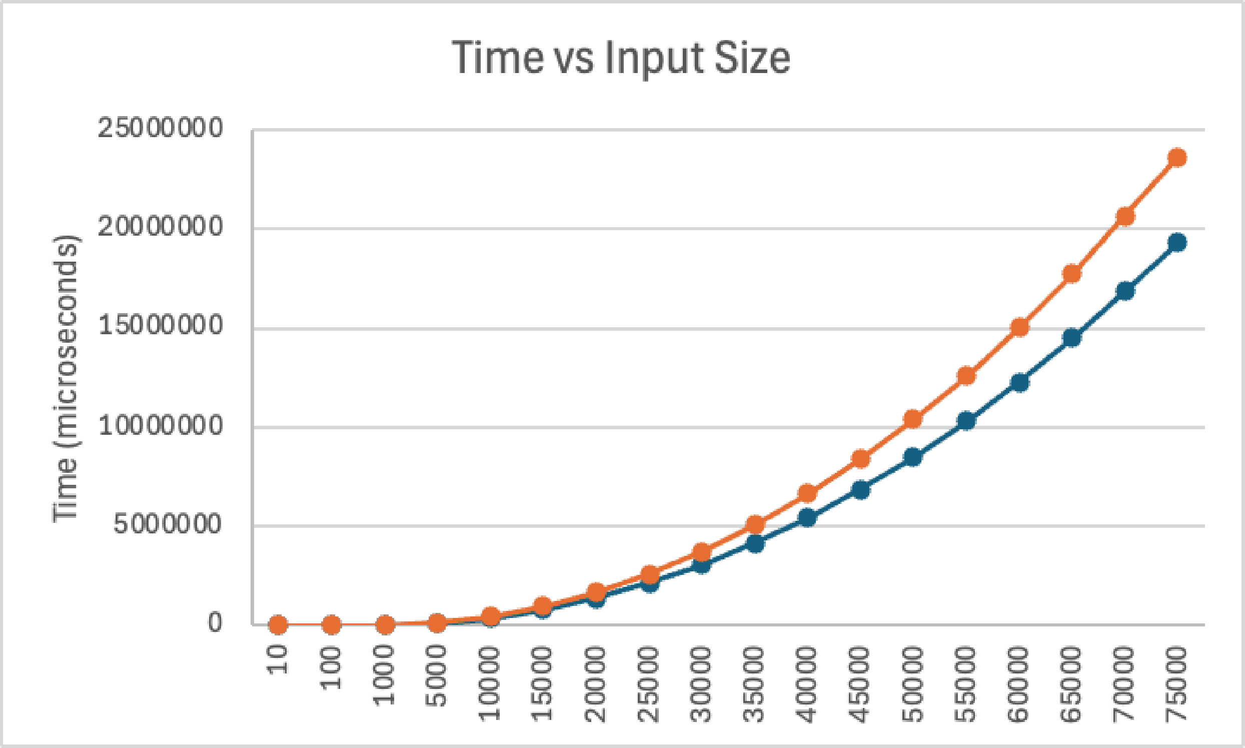
- Average & Best Cases O(n*logn)
- Worst Case O(n^2)

Graphs:

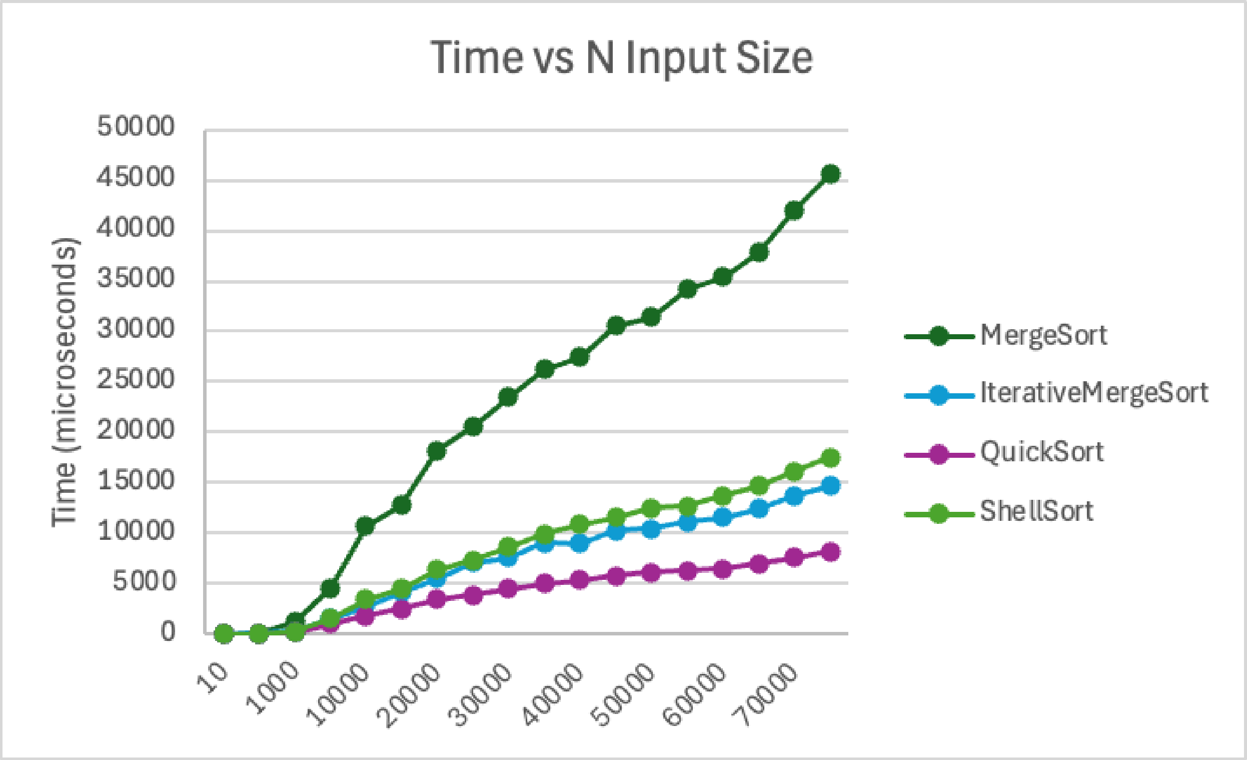
Overall graph



O(n^2) Algorithms



O(nlogn) Algorithms



Results:

From these results, we can clearly see that bubble sort and insertion sort, which have a time complexity of O(n^2) are much more inefficient compared to the other algorithms. We see a clear exponential increase as input size increases as a function of time. Despite both being O(n^2), insertion sort always performs better than bubble sort. The results from the theoretical O(nlogn) algorithms are telling of a very efficient trend between input size and elapsed time. Overall, we see the performance is magnitudes greater than both bubble sort and insertion sort with quick sort, iterative merge sort, shell sort and finally merge sort performing best to worst respectively. Although inconsistent, we can see that aside from one data point, the time it takes to sort a vector is always slightly increasing, although not by a lot. For this reason, these four algorithms are much more efficient and preferred to bubble sort and insertion sort, with quicksort always performing the best over the input size range of 10 - 75000.