Empirical Analysis Assignment 3

Student Name: Hamza Rafiq
SAP ID: 59329

Topic: Comparison of Sorting Algorithms (Bubble, Selection, Insertion, Merge)

Introduction

In this assignment, four different sorting algorithms — Bubble Sort, Selection Sort, Insertion Sort, and Merge Sort — are tested on arrays of various sizes. The goal is to analyze their performance by measuring the time they take to sort data in both ascending (best case) and descending (worst case) orders.

Methodology

Each algorithm was implemented in Python and executed multiple times to get accurate results. We used input arrays of sizes 5, 10, 50, and 100. Each algorithm ran 7 times per input, and the average execution time (in microseconds) was recorded using Python's time.perf_counter_ns() function. Graphs were plotted to compare performance for both cases.

Results & Discussion

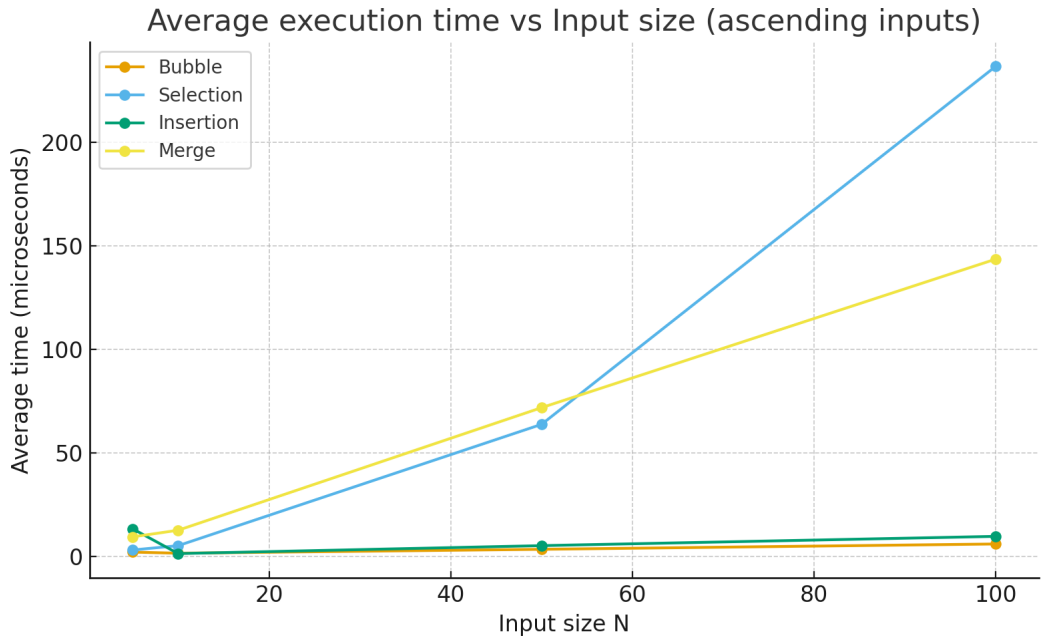
The experimental results clearly show that Merge Sort performs much faster than the other three algorithms as the input size increases. This is because Merge Sort has a time complexity of O(N log N), while Bubble, Selection, and Insertion Sorts have $O(N^2)$ complexity. In the ascending case (best case), Insertion Sort and Bubble Sort perform faster because they require fewer comparisons when the data is already sorted. However, in the descending (worst case), they become very slow due to excessive swapping. Selection Sort performs almost the same in both cases because it always scans the entire array for the smallest element in each pass, regardless of the data order. Merge Sort is stable and consistent in both cases, making it the best algorithm among all for larger inputs.

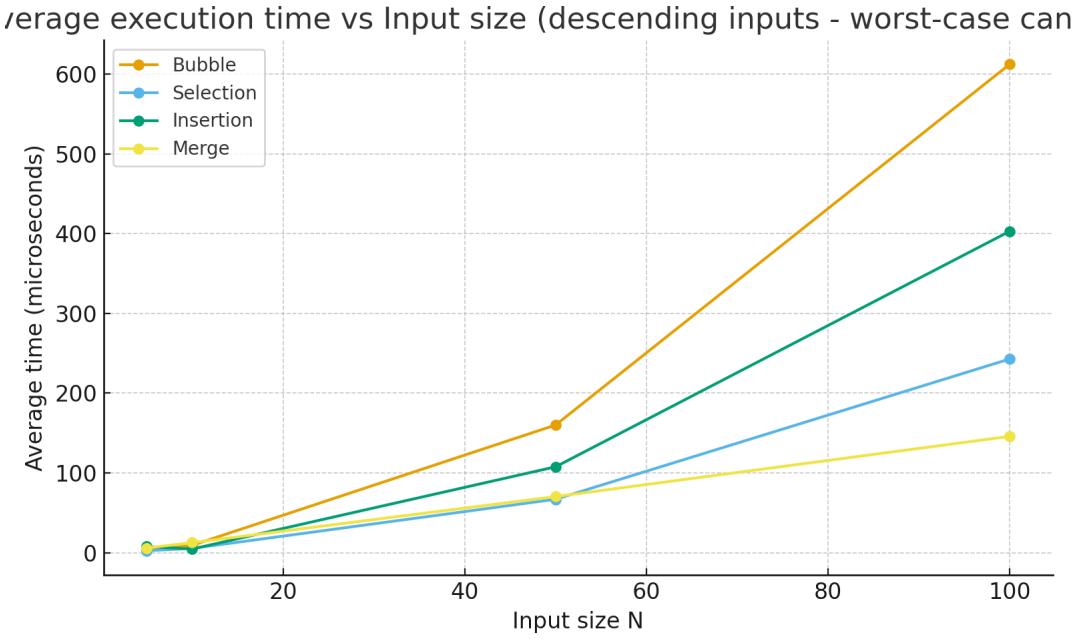
Conclusion

From the analysis, it can be concluded that Bubble, Selection, and Insertion sorts are suitable only for small data sets due to their $O(N^2)$ performance. Merge Sort is the most efficient algorithm for larger data sets, providing both speed and consistency across all input types.

Graphs & Results Summary

Below are the performance comparison graphs for ascending and descending inputs.





Prepared by: Hamza Rafiq (SAP ID: 59329)
Course: Empirical Analysis