# Analysis of Sorting Algorithm Execution Times

Omar Fayed 9035
Mohamed Khamis 9107
Yaseen Islam 9032

March 1, 2025

## 1 Introduction

Sorting algorithms play a fundamental role in computer science, affecting the efficiency of data organization and retrieval. In this experiment, we analyze the execution time of three sorting algorithms: Bubble Sort, Selection Sort, and Insertion Sort. We measure their performance on different input sizes and visualize the results.

## 2 Methodology

The execution times of Bubble Sort, Selection Sort, and Insertion Sort were measured using Python. The steps followed are:

1. Generate random arrays of different sizes.
2. Sort the arrays using the three algorithms.
3. Measure the execution time for each sorting algorithm.
4. Visualize and analyze the results.

The input sizes used were $N = [1000, 5000, 10000, 15000, 20000]$, and the execution time was recorded in milliseconds.

## 3 Results

Table 1 shows the execution time of each algorithm for different input sizes.

Figure 1 provides a visual comparison of execution times.

| Input Size (N) | Bubble Sort (ms) | Selection Sort (ms) | Insertion Sort (ms) |
|---|---|---|---|
| 1000 | 103.99 | 42.71 | 45.72 |
| 5000 | 2846.04 | 1137.61 | 1100.95 |
| 10000 | 10903.40 | 4573.44 | 4433.23 |
| 15000 | 25406.89 | 10507.70 | 10078.31 |
| 20000 | 44896.37 | 19314.61 | 18202.60 |

Table 1: Execution Times of Sorting Algorithms

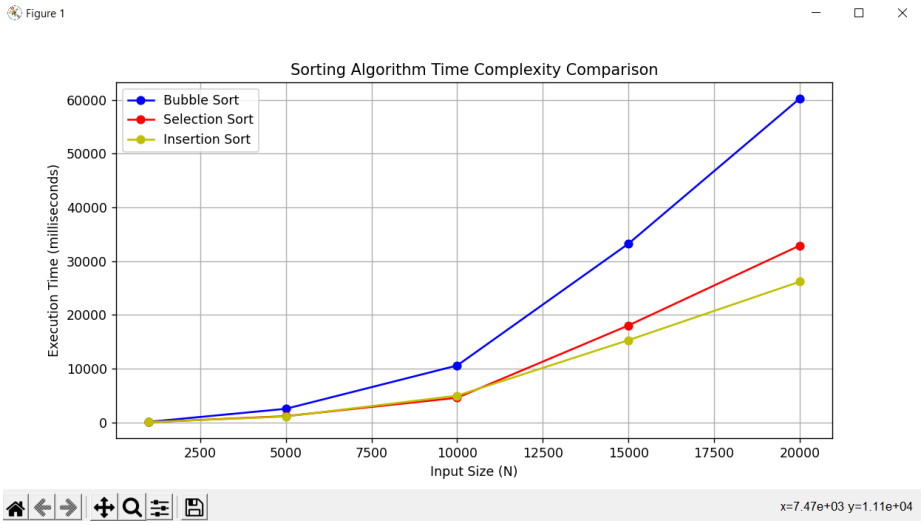


Figure 1: Execution Time Comparison of Sorting Algorithms

# 4 Conclusion

The results confirm that Bubble Sort is the slowest of the three algorithms due to its $O(N^2)$ complexity. Selection Sort also has $O(N^2)$ complexity but performs slightly better. Insertion Sort, while still $O(N^2)$, performs best for nearly sorted data. This analysis highlights the importance of choosing an efficient sorting algorithm based on the dataset size and structure.