Name: M Rehan Mehdi

Sap ID: 55189

Program: Computer Science

Semester / Section: 4/1

Course: Analysis of Algorithm

Instructor: Mr. Muhammad Usman Sharif

Assignment No.4

Table of Contents

1. Introduction

Bubble Sort is a basic sorting method. It goes through the list, compares two numbers at a time, and swaps them if they are in the wrong order. It keeps doing this until the whole list is sorted. The time it takes Bubble Sort to finish grows with the square of the number of elements ($O(n^2)$). In this assignment, I used Bubble Sort on four arrays with 5, 10, 50, and 100 numbers to see how the time changes as the list gets bigger.

2. Methodology

- I measured the time using the chrono library in C++.
- For each array size, I ran the Bubble Sort five times.
- Each time, I recorded how long it took in microseconds.
- Then, I calculated the average time to make the results more accurate.

3. Results & Graph:

- **Output of Code:** Figure 1 show output of the code:

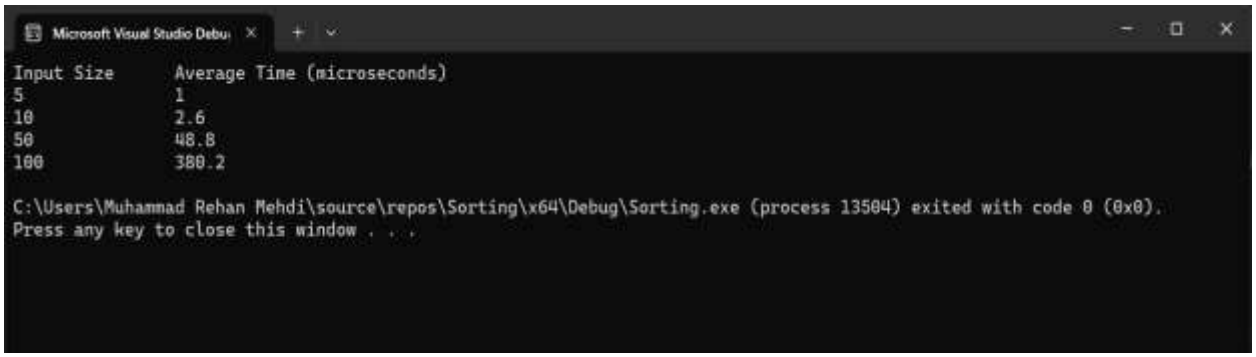
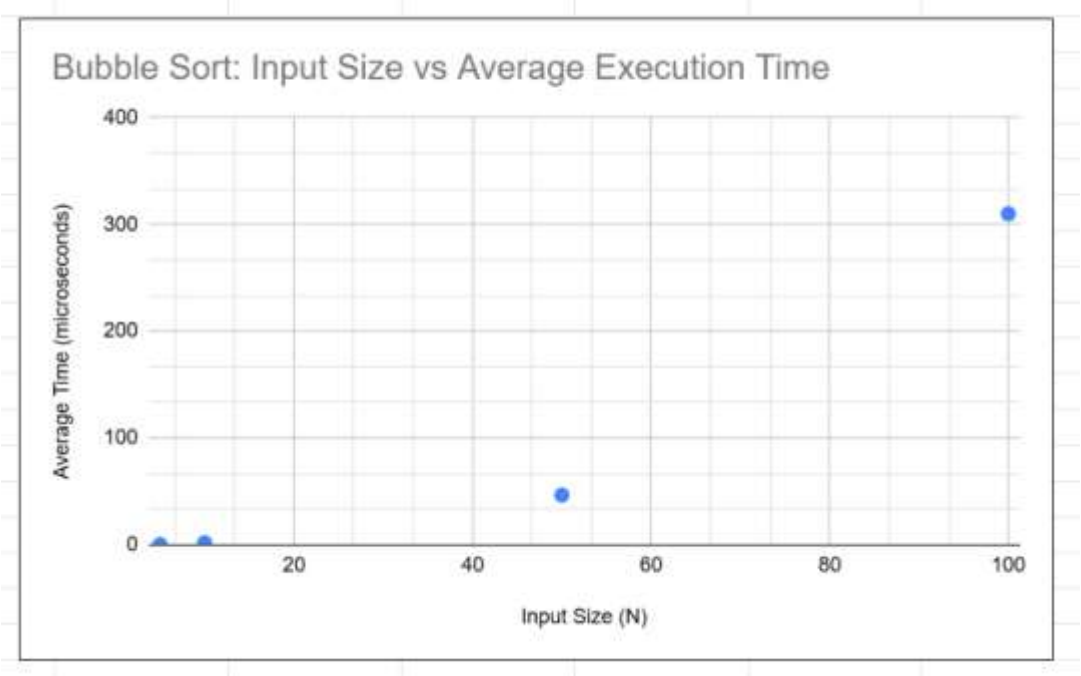


Figure 1

Graph Description:

The graph plots input size on the X-axis and average execution time (microseconds) on the Y-axis.

Graph 1 shows that as input size increases, the time grows quadratically, matching Bubble Sort's $O(n^2)$ complexity.



Graph 1

4. Analysis

- The results match the expected theoretical time complexity of Bubble Sort, which is **$O(n^2)$**.
- As the input size increases, the execution time grows much faster.
- When the number of elements doubled, the time roughly became four times more, fitting the **$O(n^2)$** pattern.
- No major problems or strange results were observed.
- Small timing differences may happen due to other programs running in the background, but overall, the results were consistent.

5. GitHub Repository Link:

https://github.com/MRehanMehdi/bubble-sort-timing.git