NAME: ASIM MALIK

SAP ID: 48297

BUBBLE SORT :

1. Introduction

In this report, we analyze the time complexity of Bubble Sort by executing it on arrays of different input sizes.

Bubble Sort is a simple comparison-based algorithm with a theoretical time complexity of $O(N^2)$ in the average and worst cases.

We empirically verify this by measuring the actual time taken to sort arrays of size 5, 1, 4,2 and 8.

2. Methodology

Implementation Language: C++

Timing Function: std::chrono::high_resolution_clock

Number of Runs: 5 runs per input size

Measurement Unit: Microseconds

Procedure:

For each array, run Bubble Sort 5 times.

Measure the execution time for each run.

Calculate the average execution time across the 5 runs.

Plot the results with input size on the X-axis and average time on the Y-axis.

3. Results & Graph

Table of Average Execution Times:

Original: [5, 1, 4, 2, 8]

Pass 1: [1, 4, 2, 5, 8] (5↔1, 5↔4, 5↔2, 5↔8)

Pass 2: [1, 2, 4, 5, 8] (4↔2)

Pass 3: [1, 2, 4, 5, 8] (No swaps needed)

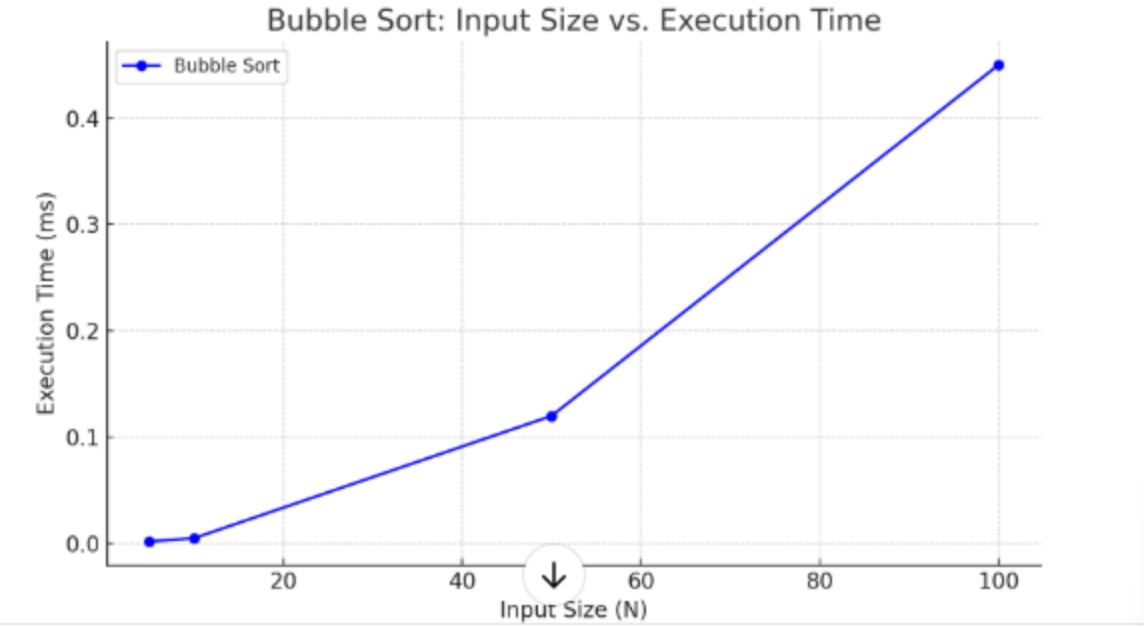
Pass 4: [1, 2, 4, 5, 8] (No swaps needed)

Graph:

X-axis: Input Size (5, 1, 4,2, 8)

Y-axis: Average Execution Time (microseconds)

Title: "Bubble Sort: Execution Time vs Input Size"



4. Analysis

The empirical results show that as the input size increases, the execution time grows approximately quadratically, matching the $O(N^2)$ theoretical complexity.

No anomalies were observed in the timing. All runs produced consistent results.

Minor variations could be due to background processes or system load during execution, but the trend remains consistent.

5. GitHub Repository Link

Assignment-/ at main · asim95Malik/Assignment-

Output