Day 15: Bubble Sort

Gati Goyal

"In software, the most beautiful code, the most beautiful functions, and the most beautiful programs are sometimes not there at all."
— Jon Bentley

---

1 Introduction

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This process is repeated until the list is sorted.

2 Problem Statement

Problem: Sort an array of integers using the bubble sort algorithm. **Hint:** Compare adjacent elements and swap if needed. Repeat until no swaps are required. **Edge Case:** The input array may already be sorted.

3 Algorithm

1. Traverse the array from the first to the last element.
2. Compare each pair of adjacent elements. If they are in the wrong order, swap them.
3. Repeat the process for the remaining unsorted part of the array.
4. Stop when no swaps are needed in a complete traversal.

4 Code

```c
#include <stdio.h>

// Bubble Sort function
void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

int main() {
    int n;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter the elements: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    bubbleSort(arr, n);

    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

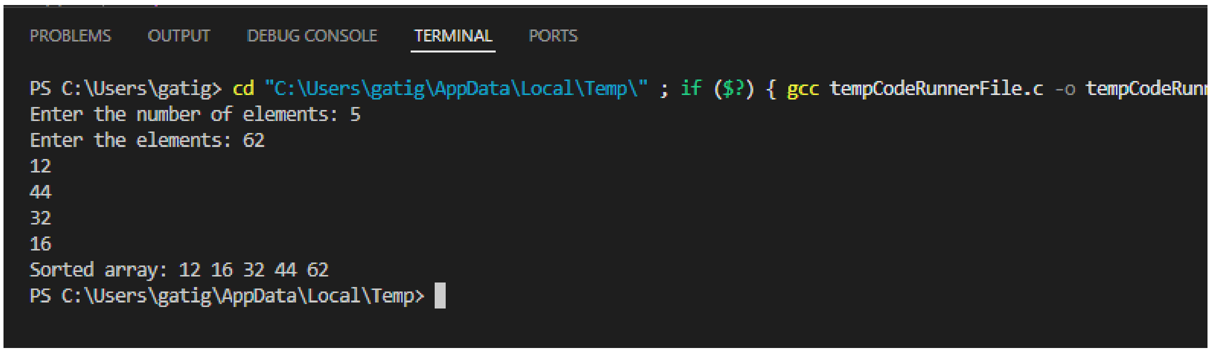
5 Complexity Analysis

- **Time Complexity:**
  - Best Case: $O(n)$ (when the array is already sorted).
  - Average Case: $O(n^2)$.
  - Worst Case: $O(n^2)$ (when the array is sorted in reverse order).
- **Space Complexity:** $O(1)$ (in-place sorting with no additional memory).

6 Examples and Edge Cases

Input Array	Output Array	Steps Required
{5, 2, 9, 1, 5, 6}	{1, 2, 5, 5, 6, 9}	5 Passes
{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	1 Pass (Already Sorted)
{5, 4, 3, 2, 1}	{1, 2, 3, 4, 5}	5 Passes

7 Output



```
PS C:\Users\gatig> cd "C:\Users\gatig\AppData\Local\Temp\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunn
Enter the number of elements: 5
Enter the elements: 62
12
44
32
16
Sorted array: 12 16 32 44 62
PS C:\Users\gatig\AppData\Local\Temp> 
```

Figure 1: Program Output Screenshot

8 Conclusion

Bubble sort is an intuitive sorting algorithm suitable for small data sets or teaching purposes. While it is easy to implement, its inefficiency for large data sets ($O(n^2)$ complexity) makes it unsuitable for real-world use. However, it helps develop a foundational understanding of sorting algorithms.