7. IMPLEMENTATION OF BUBBLE SORT

Preamble

Bubble sort is a simple and intuitive sorting algorithm. It repeatedly swaps adjacent elements if they are in the wrong order until the array is sorted. In this algorithm, the largest element "bubbles up" to the end of the array in each iteration. Bubble sort is inefficient for large data sets, but it is useful for educational purposes and small data sets. Below is the implementation of the bubble sort algorithm in C programming language.

Steps

- Start with an unsorted list of values.
- Compare the first value with the next one. If the first value is bigger, swap their positions.
- Move to the next pair of values and repeat the comparison and swapping.
- Continue until the end of the list is reached. This is one pass.
- Go back to the start of the list and repeat the steps until no more swaps are needed. The list is now sorted.

Implementation in C

```
#include <stdio.h>

void bubble_sort(int arr[], int n)

{

    int i, j;

    for (i = 0; i < n - 1; i++)

    {

        for (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 arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr) / sizeof(arr[0]);
    bubble_sort(arr, n);
    printf("Sorted array: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

Sample Input and Output

```
Sorted array: 11 12 22 25 34 64 90
```