Bubble Sort:

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. This algorithm is not suitable for large data sets as its average and worst-case time complexity is quite high.

Algorithm:

1. traverse from left and compare adjacent elements and the higher one is placed at right side.
2. In this way, the largest element is moved to the rightmost end at first.
3. This process is then continued to find the second largest and place it and so on until the data is sorted.

Code:

```
#include <bits/stdc++.h>
using namespace std;

void bubbleSort(int arr[], int n)
{
	int i, j;
	bool swapped;
	for (i = 0; i < n - 1; i++) {
		swapped = false;
		for (j = 0; j < n - i - 1; j++) {
			if (arr[j] > arr[j + 1]) {
				swap(arr[j], arr[j + 1]);
				swapped = true;
			}
		}

		// If no two elements were swapped
		// by inner loop, then break
		if (swapped == false)
			break;
	}
}

// Function to print an array
```

```
void printArray(int arr[], int size)
{
        int i;
        for (i = 0; i < size; i++)
                cout << " " << arr[i];
}

// Driver program to test above functions
int main()
{
        int arr[] = { 64, 34, 25, 12, 22, 11, 90 };
        int N = sizeof(arr) / sizeof(arr[0]);
        bubbleSort(arr, N);
        cout << "Sorted array: \n";
        printArray(arr, N);
        return 0;
}
```

Output:

```
PS D:\Mirror\ICT\3rd YEAR\SEM 5\Design and Analysis of Algorithm 01CT0512\Lab C++>  & 'c:\Users\shash\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debug
Adapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-dxttu5zl.00m' '--stdout=Microsoft-MIEngine-Out-ukrehcgp.44h' '--stderr=Microsoft-MIEngine-Er
ror-vgnrqvrv.3ts' '--pid=Microsoft-MIEngine-Pid-xewiyr15.nfg' '--dbgExe=D:\Mirror\Installations\mysys2\ucrt64\bin\gdb.exe' '--interpreter=mi'
Original array: 64 34 25 12 22 11 90
Sorted array: 11 12 22 25 34 64 90
```

Space complexity: _____

Justification:_____
_____

Time complexity:

Best case time complexity: _____

Justification:_____
_____

Worst case time complexity: _____

Justification:_____