Data Structures & Algorithms – Lab #5

Aim: Getting Familiar with Sorting Algorithms, Bubble Sort Algorithm

Topics:

1. Sorting Algorithms
2. Comparison-based Sorting Algorithms
3. Implementation of Bubble Sorting Algorithm

Lab Question

Q1 – Implement the **Bubble Sort** algorithm

(**Bubble Sort** is a comparison-based sorting algorithm that repeatedly compares pair of elements and swap them if they are not sorted. Before defining **Bubble Sort** algorithm, we will define **swap** function. The **Swap** function implements the process of swapping two elements in a list or array, having their indexes. Later in the **bubble sort** algorithm, if a pair of elements is not sorted, we can call **swap** function to swap them and make that pair sorted.)

swap Function (Swapping two elements in data list by having their indexes, i and j)

```
####### Swapping two elements ########

def swap(data, i, j) :
    temp = data[i]
    data[i] = data[j]
    data[j] = temp
```

Bubble Sort Function

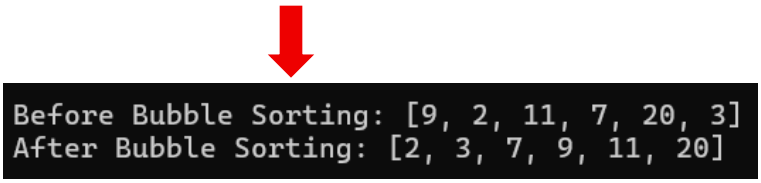
```
######## Bubble Sort Function #########

def bubbleSort(data) :
    swapped = True
    last = len(data) - 1
    while(swapped) :
        swapped = False
        for i in range(0, last) :
            if data[i] > data[i+1] :
                swap(data, i, i+1)
                swapped = True
```

Calling bubble sort function on a sample data:

```
data = [9,2,11,7,20,3]
print("Before Bubble Sorting:", data)

bubbleSort(data)
print("After Bubble Sorting:", data)
```



```
Before Bubble Sorting: [9, 2, 11, 7, 20, 3]
After Bubble Sorting: [2, 3, 7, 9, 11, 20]
```