Data Structures & Algorithms – Lab #6

Aim: Getting Familiar with Sorting Algorithms, Insertion Sort Algorithm

Topics:

1. Sorting Algorithms
2. Implementation of Insertion Sorting Algorithm
3. Comparing Execution Time of Insertion Sort and Bubble Sort

Lab Question

Q1 – Implement the **Insertion Sort Algorithm**, then do the following:

- Declare and initialize the list below:

  sample1 = [4, 9, 2, 10, 3, 7, 5]

- Sort the **sample1** list by calling the algorithm.

Insertion Sort Function

```
######### Insertion Sort Function #########

def insertionSort(data):
    n = len(data)
    for i in range(1,n):
        temp = data[i]
        j = i
        while (j>0) and (data[j-1]> temp):
            data[j]=data[j-1]
            j-=1
        data[j]=temp
```

Calling the insertion sort function on a sample data:

```
sample1 = [4, 9, 2, 10, 3, 7, 5]
print('Before Insertion Sort:', sample1)

insertionSort(sample1)
print('After Insertion Sort:', sample1)
```

```
Before Insertion Sort: [4, 9, 2, 10, 3, 7, 5]
After Insertion Sort: [2, 3, 4, 5, 7, 9, 10]
```

Q2 – Write the required Python code to do the following:

- Generate a list of 1000 random integer numbers between 1 and 100. (list name → **sample2**)
- Sort the list of unsorted random numbers by calling the insertion sort algorithm.
- Calculate the execution time of sorting done by the algorithm.

```
import random
import time

sample2 = []

#### Generating 1000 random integer number between 1 and 100 ####
for i in range(1000):
    num = random.randint(1,100)
    sample2.append(num)

######## Calculating Execution Time of Insertion Sort #########
start = time.time()
insertionSort(sample2)
end = time.time()

print('Execution Time of Insertion Sort Algorithm:', end-start)
```

Q3 – Repeat the previous example by implementing the **Bubble Sort Algorithm**. Which one is faster in sorting the unsorted list? Insertion sort or Bubble sort?

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

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

```
######## 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
```

```
import random
import time

sample2 = []

#### Generating 1000 random integer number between 1 and 100 ####
for i in range(1000):
    num = random.randint(1,100)
    sample2.append(num)

######## Calculating Execution Time of Bubble Sort #########
start = time.time()
bubbleSort(sample2)
end = time.time()

print('Execution Time of Bubble Sort Algorithm:', end-start)
```

```
Execution Time of Bubble Sort Algorithm: 2.7358050346374511
```

- The insertion sort algorithm is faster, sorting the list in 0.3785 seconds, while the bubble sort algorithm sorts it in 2.7358 seconds.