Data Structures & Algorithms – Lab #7

Aim: Getting Familiar with the Merge Sort Algorithm, Divide-and-Conquer Approach

Topics:

1. Divide-and-Conquer Approach
2. Implementation of the Merge Sorting Algorithm
3. Evaluating the Execution Time of Merge Sort

Lab Questions

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

- Declare and initialize this list → sample1 = [4, 9, 2, 10, 3]
- Sort the **sample1** list by calling the algorithm.

Merge Sort Function

```
################## Merge Sort Function ##################
def mergeSort(data):
    # Base Case
    if len(data)<=1:
        return data

    # Dividing the dataset to Two Halves (Left and Right)
    mid = len(data)//2
    left = data[:mid]
    right = data[mid:]

    # Recursion (Calling the algorithm for each half)
    mergeSort(left)
    mergeSort(right)

    # Merging Two Halves
    i = 0
    j = 0
    k = 0
    while i<len(left) and j<len(right):
        if left[i]<= right[j]:
            data[k] = left[i]
            i += 1
        else:
            data[k] = right[j]
            j += 1
        k += 1

    # Handling the Remaining Items in Both Halves
    while i<len(left):
        data[k] = left[i]
        i += 1
        k += 1
    while j<len(right):
        data[k] = right[j]
        j += 1
        k += 1
    return data
```

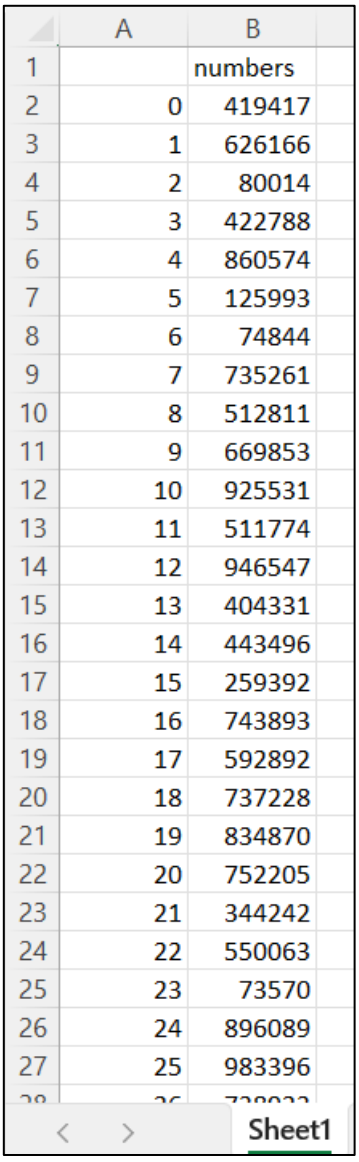
Calling the merge sort function on a sample data:

```
sample1 = [4, 9, 2, 10, 3]
print("Before Merge Sort:", sample1)

mergeSort(sample1)
print("After Merge Sort:", sample1)
```

Q2 – Call the **Merge sort algorithm** to sort a list of 1000 random integer numbers between 1 and 1000000, and calculate its execution time:

- Read the numbers from an Excel file, named "test.xlsx".
- "test.xlsx" has one column named "numbers" containing the list of 1000 random numbers.



	A	B
1		numbers
2	0	419417
3	1	626166
4	2	80014
5	3	422788
6	4	860574
7	5	125993
8	6	74844
9	7	735261
10	8	512811
11	9	669853
12	10	925531
13	11	511774
14	12	946547
15	13	404331
16	14	443496
17	15	259392
18	16	743893
19	17	592892
20	18	737228
21	19	834870
22	20	752205
23	21	344242
24	22	550063
25	23	73570
26	24	896089
27	25	983396

Sheet1

```
# pandas library is used to read an excel file data in Python IDE
import pandas as pd
import time

# read_excel() is a method in pandas libarary to read the excel file
data = pd.read_excel(r"File Path of test.xlsx")

# Extracting "numbers" column values and converting it to a list
sample2 = data['numbers'].tolist()

# Calling merge sort algorithm and calculating its execution time
start = time.time()
mergeSort(sample2)
end = time.time()

print("Execution Time of Merge Sort Algorithm:", end-start)
```

```
Execution Time of Merge Sort Algorithm: 0.04762458801269531
```