UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

Worksheet 2

Student Name: Abhishek Kumar **UID: 24MCA20431**
Branch: MCA **Section/Group: 6 (B)**
Semester: II **Date of Performance: 28/01/25**
Subject Name: DAA LAB **Subject Code: 24CAP-612**

1. **Aim/Overview of the practical:** Implement merge sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator

2. **Task to be done:**

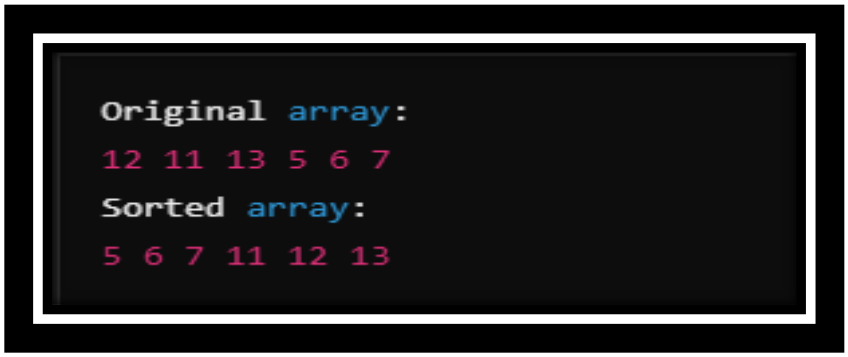
   Implement the merge sort algorithm, measure its execution time for different input sizes, and plot a graph of time taken versus the number of elements.

3. **Code for experiment/practical:**

```
public static void mergeSort(int[] array, int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;

        mergeSort(array, left, mid);
        mergeSort(array, mid + 1, right);

        merge(array, left, mid, right);
    }
}

public static void merge(int[] array, int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;

    int[] leftArray = new int[n1];
    int[] rightArray = new int[n2];
```

```
    for (int i = 0; i < n1; i++)
        leftArray[i] = array[left + i];
    for (int j = 0; j < n2; j++)
        rightArray[j] = array[mid + 1 + j];

    int i = 0, j = 0, k = left;

    while (i < n1 && j < n2) {
        if (leftArray[i] <= rightArray[j]) {
            array[k] = leftArray[i];
            i++;
        } else {
            array[k] = rightArray[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
        array[k] = leftArray[i];
        i++;
        k++;
    }

    while (j < n2) {
        array[k] = rightArray[j];
        j++;
        k++;
    }
}

public static void printArray(int[] array) {
    for (int num : array) {
        System.out.print(num + " ");
    }
    System.out.println();
}
```

4. **Result/Output/Writing Summary:**



Learning outcomes (What I have learnt):

1. **Learned how to implement merge sort in a programming language.**
2. **Observed how sorting time varies with values of n.**
3. **Gained knowledge of the merge sort algorithm and its divide-and-conquer approach.**

Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Pre Lab Quiz)		5
2.	Worksheet		10
3.	Post Lab Quiz		5