B.Sc. Engg. CSE 8th Semester **February 2023**

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

LAB 3 **Summer 2021-2022**

CSE 4810: Algorithm Engineering Lab

This lab is focused around exploring Divide and Conquer algorithms.

Reference: Introduction to Algorithms, 3rd Edition, Thomas. H. Cormen *et al.*

1. In the lecture, we explored merge sort and how divide and conquer technique helps in solving the problem of sorting an array in $\boldsymbol{\theta(nlogn)}$ time. Simulate how this sort would work for an array containing the numbers: 8,3,10,7,9,4,0,2,6,1.

2. The way *Divide and Conquer* problems are formulated (defining sub-problems, solving those subproblems and combining them to create a solution to the bigger problem) is suspiciously like the category of problems to which Dynamic-programming is applicable. If a problem can be solved using a Divide and Conquer technique, is Dynamic Programming guaranteed to be applicable to it?

3. Imagine you are given k sorted arrays. All these arrays are of size N. Your task is to merge them into a single sorted array in **O(N*k*log(k))** time complexity. Design an algorithm to accomplish this.

4. Write the algorithm to find a count array given an integer array where the count array contains count of how many smaller elements to the right of any element there are. Analyze the complexity of your solution. Can you devise a faster solution.

```
Input: nums = [5,2,6,1]
Output: [2,1,1,0]
Explanation:
To the right of 5 there are 2 smaller elements (2 and 1).
To the right of 2 there is only 1 smaller element (1).
To the right of 6 there is 1 smaller element (1).
To the right of 1 there is 0 smaller element.
```

5. You are given two sorted arrays of size N and M. Write an algorithm that finds out the median of their elements combined. First, devise an algorithm with any complexity. Then, optimize it and find out an algorithm that can solve the problem in O(log (N+M)) time complexity.

```
Input: nums1 = [1,3], nums2 = [2]
Output: 2.00000
Explanation: merged array = [1,2,3] and median is 2.
```