**National University of Computer & Emerging Sciences (NUCES) Islamabad,**

Department of Computer Science

**DATA STRUCTURES – FALL 2023**

**LAB 04**



**Learning Outcomes**

In this lab you are expected to learn the following:

* Sorting and Searching Algorithms
* Multidimensional Arrays

**Objective**

To implement various sorting and searching algorithms and apply them on multidimensional arrays.

# Task 1(50 minutes):

Eric loves puzzle making. One night he went to sleep after completing his puzzle only to wake up and find his puzzle had been messed up. Help Eric in **sorting** the puzzle pieces back to correct form.

* The first row of puzzle has to be sorted using **insertion sort**
* The second row of puzzle has to be sorted using **selection sort**
* The third row of puzzle has to be sorted using **bubble sort**

**Unsorted Puzzle**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 32 | 30 | 39 | 34 | 37 | 35 | 33 |  |  |  |  |  |
| 50 | 48 | 54 | 59 | 47 | 49 | 52 | 51 | 55 | 53 | 57 | 58 |
| 9 | 3 | 6 | 5 | 12 |  |  |  |  |  |  |  |

**Sorted Puzzle**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 30 | 32 | 33 | 34 | 35 | 37 | 39 |  |  |  |  |  |
| 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 57 | 58 | 59 |
| 3 | 5 | 6 | 9 | 12 |  |  |  |  |  |  |  |

**Note: You can create a separate class in which you implement the required sorting algorithms.**

# Task 2(40):

You are given a crossword puzzle in the form of 2D-Array.

|  |  |  |
| --- | --- | --- |
| C | E | M |
| B | A | O |
| X | W | T |

Your task is to find the indices of the characters of the word: **CAT.**

Let us walk through this example:

You have a crossword puzzle (2D-character Array) and the word ( 1D- character array).

While you traverse the crossword puzzle you have to **search(linear)** whether the character at the current index is a character from the word. If yes, then store the row index and column index of that particular character. In the end you have to return a 2D- array of indices of the said characters. The first row of resultant 2D-array should represent the row indices and the second row should represent the column indices.

**Output:**

**row index of ‘C’ row index of ‘A’ row index of ‘T’**

|  |  |  |
| --- | --- | --- |
| **0** | **1** | **2** |
| **0** | **1** | **2** |

**column index of ‘C’ column index of ‘A’ column index of ‘T’**

# 

# Task 3(30):

A **Markov Matrix** is the one in which the sum of each row is equal to 1. You are required to implement a function that checks whether a matrix is Markov matrix or not.

Markov Matrix

|  |  |  |
| --- | --- | --- |
| **0.1** | **0.6** | **0.3** |
| **0.3** | **0.4** | **0.3** |
| **0.5** | **0.0** | **0.5** |

Not a Markov Matrix

|  |  |  |
| --- | --- | --- |
| **0.1** | **0.9** | **0.3** |
| **0.5** | **0.4** | **0.3** |
| **0.5** | **0.9** | **0.5** |