CSE 201 Data Structures – Homework 01

Due Date: 23:59 Sunday September 29th

Disclosure

You will submit your file to an assignment that is given through MS teams. Your filename should be *HW01_yourStudentNumber.java*. You are not required to provide a main method, however you can write one to test your code. Submissions made after the deadline will not be accepted, be sure to submit your work before the due date. Your code will be automatically graded and checked for plagiarism, so be sure to have the same class, method, variable names. Failure to do so may result in you receiving 0 from this exercise. All classes should be written to a single Java file.

Objective

In This assignment, you will implement a simplified traffic control system using a 2D array to represent the state of traffic lights in a city. The goal is to practice manipulating and acessing data stored in 2D arrays and using loops, conditionals, and basic data structures.

Problem

You are tasked with designing a program to monitor traffic lights in a grid-shaped city. Each traffic light can be in one of three states:

- 0: Light is functioning normally.
- 1: Light is malfunctioning.
- 2: Light is under maintenance.

You will be implementing *TrafficControl* class according to given *ITrafficControl* interface, which models the city's traffic light system as a 2D grid. Your task is to help city officials monitor malfunctioning lights and identify crucial intersections where multiple lights are malfunctioning. Given interface:

```java
interface ITrafficControl {
    int[][] getCityGrid();
    int countMalfunctioningLights();
    String mostMalfunction();
    int countMalfunctioningNeighbors(int row, int col);
    List<int[]> crucialIntersections();
    String toString();
}
```

In addition to given interface, your class must have 2 constructors. First constructor accepts the parameters of the grid and randomly assigns 0, 1, or 2 to generate a grid:

```java
public TrafficControl(int m, int n)
```

Second constructor, takes the grid:

```
public TrafficControl(int[][] cityGrid)
```

Task

1. **Get City Grid:** Method that returns the grid.
2. **Count Malfunctioning Lights:** Write a method *countMalfunctioningLights()* that returns the number of malfunctioning traffic lights (lights in state 1).
3. **Identify Problematic Rows or Columns:** Write a method *mostMalfunction()* that identifies which row or column in the city grid has the highest number of malfunctioning traffic lights. The method should return either the row or column index depending on which has more malfunctioning lights. Example output: “Row: 2”, or “Column: 1”.
4. **Count Neighboring Malfunctioning Lights:** Write a method *countMalfunctioningNeighbors(int row, int col)* that takes a specific intersection (row and column) and counts how many of its immediate neighbors (up, down, left, right) are malfunctioning
5. **Find Crucial Intersections:** Write a method *crucialIntersections()* that returns a list of all intersetions (represented as (row, column) coordinates) that have two or more malfunctioning neighbors.
6. **City Representation:** Override the *toString()* method that returns a string representation of the 2D city grid, where each cell in the grid represents the current state of a traffic light.
Example:

```
2   1   2   2   2

0   0   1   1   0

1   1   0   0   1

0   0   0   0   2
```