CS310 Python for Data Science

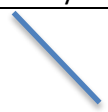
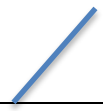
Assignment 2

In this assignment, you will implement a function to track the path of light ray inside a mirror maze.

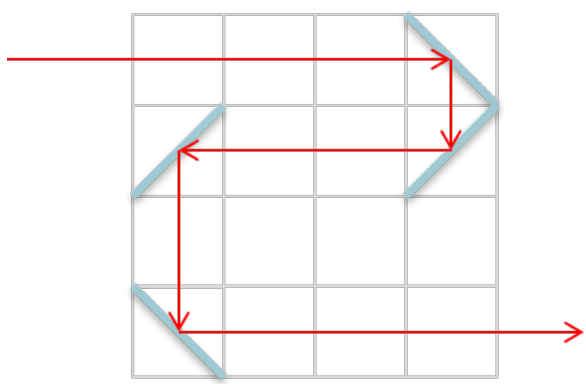
The mirror maze will be inputted as a 2D list argument. For example:

```
maze = [[ 0,  0,  0, -1],
        [ 1,  0,  0,  1],
        [ 0,  0,  0,  0],
        [-1,  0,  0,  0]]
```

represents a mirror maze in 4x4 space. The space is virtually divided into 16 cells. Each cell contains a number, which represents the configuration in that cell:

0	Empty cell, light ray can travel freely.
-1	 A two-sided mirror installed in -45 degrees angle
1	 A two-sided mirror installed in 45 degrees angle.

The above 2D list maze example can be visualized as the following maze:



Assuming the light ray always comes from the left side horizontally entering the top-left square. In this specific example, the light ray will hit those four mirrors and change its direction in this maze, until it goes out of the maze.

Your function **maze_solver** should take that 2D list as the input maze and output the path of the light ray travels in this maze. For this example, the travel path is:

[(0, 0), (0, 1), (0, 2), (0, 3), (1, 3), (1, 2), (1, 1), (1, 0), (2, 0), (3, 0), (3, 1), (3, 2), (3, 3)]

where each **tuple** represents the cell's row and column index.

Your function should be called like this sample run:

```
Path = maze_solver(maze)
```

A few more notes:

1. You can always assume the input maze is valid.
2. the maze is a 2D grid with arbitrary number of columns and rows.
3. The max size of the maze will be at 100x100.
4. Each cell can only contain 0, 1, or -1.
5. It is impossible to create a circular path for the light ray to loop forever inside the maze. No need to worry about that scenario.
6. Light ray always comes from the left side horizontally to enter the (0, 0) cell.
7. The light ray only travels horizontally or vertically because all the mirrors are installed at +-45 degrees.
8. It is possible the light ray may pass the same cell multiple times.
9. You should be able to implement using only what we learned in class so far.
10. You are encouraged to discuss with other students about the high-level ideas, but never share anything at code level. You will have to write your own code independently. **Plagiarism will be detected and reported.**

Deliverable:

You will submit a single python (*.py) file using the Canvas assignment page before the due time. Other file types will not be accepted.

How do I grade:

I will use 10 test cases to test your function. Each test case will be 10 points. No partial points within each test case.

I will post the test cases after the due date. So please test thoroughly before you submit.

Total 100 points.