

**Q-1:** Write a function to detect cycles in a directed graph using DFS traversal. Return true if the graph contains a cycle; otherwise, return false. (10 Points)

**Q-2:** Given a **snake and ladder board**, find the minimum number of dice throws required to reach the destination or last cell from the source or 1st cell. Basically, the player has total control over the outcome of the dice throw and wants to find out the minimum number of throws required to reach the last cell.

If the player reaches a cell which is the base of a ladder, the player has to climb up that ladder and if reaches a cell is the mouth of the snake, and has to go down to the tail of the snake without a dice throw. (10 Points)

**Q-3:** Given a two-dimensional grid, each cell of which contains an integer cost which represents a cost to traverse through that cell, we need to find a path from the top left cell to the bottom right cell by which the total cost incurred is minimum. (10 Points)

**Note:** It is assumed that negative cost cycles do not exist in input matrix.

**Q-4 :**

You are given a grid of size “ n x m” . Grid contains 0's and 1's. Where 0's means water area and 1's means land area. Find number of islands. Where island is formed by connecting adjacent lands diagonally, vertically or horizontally.

**Q-5 :**

You are tasked with analyzing a social network represented as an undirected graph, where each node represents an individual and each edge represents a friendship connection between individuals. Your goal is to write a program that utilizes both Breadth-First Search (BFS) and Depth-First Search (DFS) algorithms to perform the following tasks:

Implement a function **find\_fof(graph, person)** that takes the social network graph and a specific individual's name (**person**) as input and returns a list of their friends-of-friends (FoF). Friends-of-friends are individuals who are connected to the given person through exactly two friendship connections.

**Note:** You may assume that the input social network graph is represented as an adjacency list or an adjacency matrix.