Lab Assignment

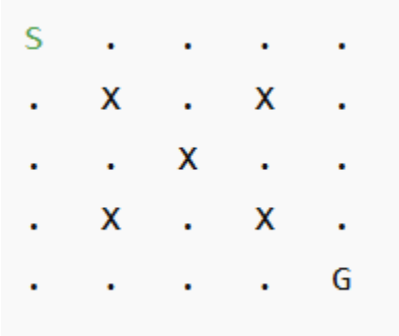
Implement **Q-Learning** to train an agent to navigate a gridworld environment and reach a goal state while avoiding obstacles.

Problem Statement

You are required to implement a **Gridworld** environment where an agent must learn an optimal policy using **Q-Learning**.

- Grid Size: 5 × 5
- Start State: (0, 0)
- Goal State: (4, 4) - reward = +100
- Obstacles (agent cannot pass through these cells): (1, 1),(2, 2),(3, 1),(3, 3),(1, 3)

Visualization:



Legend:

- S = Start
- G = Goal
- X = Obstacle
- . = Free cell

Actions (A):

- UP
- DOWN
- LEFT
- RIGHT

Transition Rules

- If an action tries to go outside the grid → agent stays in the same state
- If action hits an obstacle → agent stays in same state
- Otherwise → agent moves to the next state

Rewards

- Goal (state(4,4)) → +100
- Obstacle (if collided) → −10
- Every step (non-terminal) → −1
- Episode ends when goal is reached

Given Data:

- Learning rate α=0.1
- Discount factor γ=0.99
- Exploration rate ε=0.1
- Episodes = **1000**

Tasks to Complete

Task 1 - Implement the Gridworld environment

1. Define the 5×5 grid
2. Define allowed transitions
3. Include reward logic
4. Implement step(state, action) function

Task 2 - Implement Q-Learning

1. Initialize Q(s,a) matrix to zeros
2. Implement ε-greedy action selection
3. Apply Q-learning update rule
4. Run training for given number of episodes

Task 3 - Extract Optimal Policy

Task 4 - Plot Episode Rewards

1. Plot total reward per episode.

   Required plot:

   - X-axis: Episode number
   - Y-axis: Total reward in episode

All The Best !