Homework 3: Due on 03/07/2025
ESE 559 – Learning and Planning in Robotics

Problem 1: Consider a robot residing in the grid world environment of Fig.1. The actions of the robot are A = {up, down, left, right, idle}. The task of the robot is to eventually reach the goal location "g" with the smallest number of steps starting from "s", while avoiding obstacles (blue). The robot does not need to stay in the location "s" once it arrives there. In other words, what happens after the robot reaches "s" is irrelevant. Therefore, you can treat the blue cells and the "g" cell as terminal MDP states.

For simplicity, assume that the available actions at each state are only the ones that can keep the robot inside the workspace. When the robot chooses the action "idle" it stays with probability 1 in its current state s. For all other actions, the probability of going to the correct/intended state is "p" and the probability of going to a neighboring state is (1-p)/m(s) where m(s) is the number of neighboring states of the current state s excluding the correct state. For instance, for the initial state s shown in Fig 1, we have that (i) the only available actions are {up, right, idle} and (ii) m(s)=1, P(s,`up', s1)=p, P(s,`up', s2)=1-p

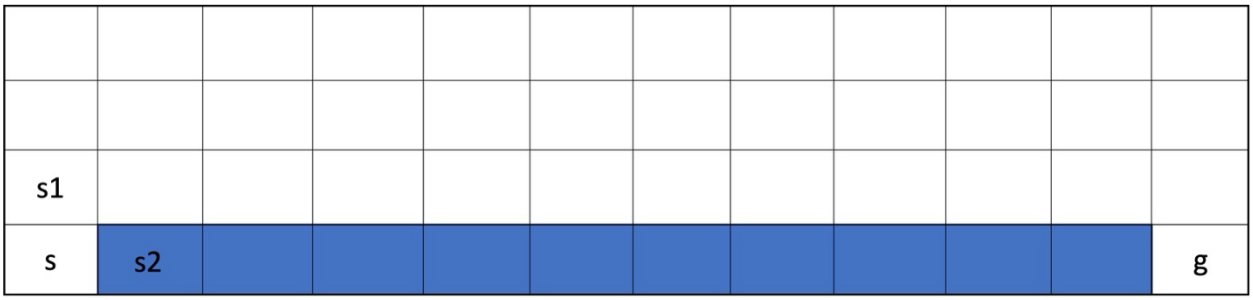

Fig 1. Environment for Problem 1

A) Implement a reward function that returns -100 every time an obstacle is hit, +100 when the goal location is reached, and -1 otherwise.

B) Implement the SARSA and Q Learning algorithms discussed in class and apply them to learn the optimal policy for p=0.65 and p=0.95. Provide figures showing (i) the optimal policy at each state and (ii) the evolution of $Q(s,\pi(s))$ over iterations for both values of p. Which algorithm seems to perform better in terms of convergence speed and variance/fluctuations of the Q value function? Does the performance seem to depend on the value of p?