Assignment 3 - Reinforcement Learning

1 Overview

For this assignment you will be making a maze solver. Your program will generate a maze of size NxN. Also you should generate barriers at random grid locations. Then you will try to learn the path out of the grid using policy and value iteration.

2 Application

2.1 Algorithms

You will need to implement the NxN maze solver problem using the Policy Iteration and Value Iteration.

Policy Iteration (PI)

1. i=0; Initialize $\pi_0(s)$ randomly for all states s
2. While i == 0 or $|\pi_i - \pi_{i-1}| > 0$ ← **Use a L1 norm: measures if the policy changed for any state**
   - Policy evaluation: Compute value of π_i
   - i=i+1
   - Policy improvement:

$$Q^{\pi_i}(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V^{\pi_i}(s')$$

$$\pi_{i+1}(s) = \arg\max_a Q^{\pi_i}(s,a)$$

Value Iteration (VI)

1. Initialize $V_0(s)=0$ for all states s
2. Set k=1
3. Loop until [finite horizon, convergence]
   - For each state s

$$V_{k+1}(s) = \max_a R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s')$$

   - View as Bellman backup on value function

$$V_{k+1} = BV_k$$
$$\pi_{k+1}(s) = \arg\max_a R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s')$$

3 Notes

The output should be in a traceable format; as every step should be printed.(best visualization will be grant a bonus).

3.1 Deliverables

1. Your well commented code.
2. A report showing your work, including:
   (a) path to goal
   (b) cost of path
   (c) running time

Also the report should contain the data structure used (if any) and algorithms, Assumptions and details you find them necessary to be clarified, Any extra work and Sample runs. You should show your algorithm and how it operates.

3.2 Further Notes

You may use Java , Python or C++ for your implementation.
Copied assignments will be severely penalized.
You can work in groups of 3.

Good Luck