Reinforcement Learning:
Coursework Assignment 1 (Semester 2, 2012)

Subramanian Ramamoorthy, Majd Hawasly

Instructions:

1. This homework assignment is to be done *individually*, without help from your classmates or others. Plaigarism will be dealt with strictly as per University policy.
2. Solve all problems and provide your complete solutions (with adequate reasoning behind each step) in a computer-printed or *legibly* handwritten form.
3. For computational questions, include your code (e.g., Matlab commands) and all major numerical parameters involved.
4. This assignment will count for 10% of your final course mark.
5. Please submit your assignment by 4 pm on 1st March - paper copy to ITO as well as an electronic version via the submit system (including code).

Questions:

1. Design and build a learning agent that operates in a grid world, specified in figure 1. The goal of the robot is on the top right hand corner, where the agent should expect to receive a reward. The grid world includes barriers which prevent certain actions from being used in corresponding cells. The objective of learning is to find policies that take the agent to the goal from all possible initial states.

   In this problem, you may assume the following:
   State: you know what square the agent is occupying at any given time.
   Action: N, S, E, W.
   State transitions: Initially, deterministic and can't pass through walls. Agent remains in the same square if it attempts a disallowed action. In a later section, we'll modify this further.
   Rewards: -1 for each time step, 0 for attaining the goal.

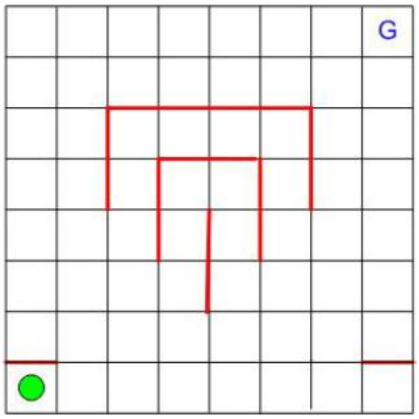

Figure 1: Grid world for Question 1. The green dot is one instance of an initial state and G denotes the goal cell.

(a) Figure 2 shows you *part* of a deterministic non-optimal policy. Complete the policy as you wish (the intention is to avoid obviously optimal actions - so complete this in a way that involves long detours). Implement policy evaluation, for V^{π}, based on dynamic programming. Give the resulting value function and the corresponding greedy policy. Also, describe the procedure in terms of key steps and any noteworthy design decisions.

(b) Perform policy iteration to get the optimal policy, still based on dynamic programming. Again, give the resulting value function and corresponding optimal policy, as well as a description of key design decisions and outline of your procedure.

(c) Perform value iteration using the same specifications as before. Compare the time taken by policy iteration and value iteration.

(d) Consider the case where the right wall is made 'sticky', i.e., from states adjoining that wall actions do not succeed (i.e., leave you in the same cell) with probability p (initially, set $p = 0.4$). Repeat the value iteration process for this system. Describe the changes to your problem formulation and depict the optimal value function and policy. Also discuss the effect of the parameter p on the optimal policy.

[50 points (10 + 10 + 15 + 15)]

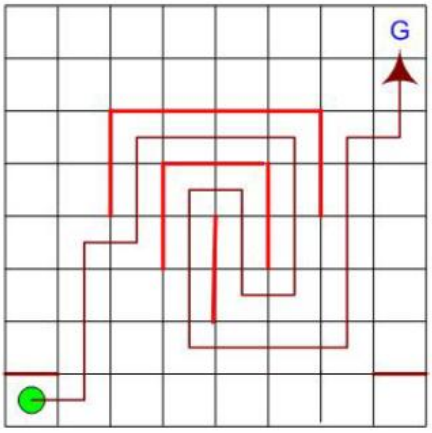

Figure 2: Part of a non-optimal policy - complete it as specified in the question.

2. Consider the secretary problem, which in its simplest form is defined as follows:

   - There is one secretarial position available.
   - The number n of applicants is known.
   - The applicants are interviewed sequentially in random order, each order being equally likely.
   - It is assumed that you can rank all applicants from best to worst without ties. The decision to accept or reject must be based solely on the relative ranking of applicants interviewed *so far*.If you reach the final applicant, you are forced to hire that person.
   - All decisions are instantaneous and final - an applicant already rejected can't be reconsidered.
   - Your decision criterion is to maximize the quality of the chosen candidate.

   (a) Pose this problem in the form of a Markov Decision Process.

   (b) Implement the on-policy Monte Carlo control algorithm for determining the optimal choice, with $n = 30$. Depict the optimal value function and policy.

(c) Implement the Q-learning algorithm for determining the same optimal choice; depict the optimal value function and policy, and comment on the difference between the Monte Carlo and Temporal Difference solutions (considering both time and quality of the solution).

[50 points (15 + 15 + 20)]