CSE 412A Spring 2022 — Introduction to Artificial Intelligence — Exercise 8

- You have approximately as many minutes as there are points.
- Mark your answers ON THE EXERCISE ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation. All short answer sections can be successfully answered in a few sentences AT MOST.
- For True/False questions, please *circle* your answer.

First name	
Last name	
WUSTL ID	

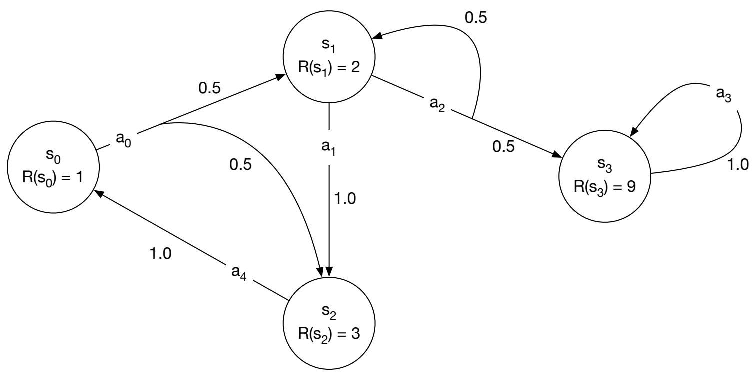
For staff use only:

Q1.	MDPs and Reinforcement Learning	/32
	Total	/32

THIS PAGE IS INTENTIONALLY LEFT BLANK

Q1. [32 pts] MDPs and Reinforcement Learning

Consider the MDP on the right with four states s_0, s_1, s_2, and s_3 and their corresponding rewards denoted in the nodes. The arrows represent the transition function with the probabilities denoted with each arrow. For example, $T(s_0, a_0, s_1) = 0.5$ and $T(s_0, a_0, s_2) = 0.5$.



(a) [10 pts] Compute the numerical value of $V_1(s_1)$ and $V_2(s_1)$, i.e., the value of state s_1 after the first two iterations of Value Iteration. Assume that the discount factor $\gamma = 0.9$ and the initial values of all states in the zero-th iteration are all zero, i.e., $V_0(s_0) = V_0(s_1) = V_0(s_2) = V_0(s_3) = 0$.

Recall that the update equation of Value Iteration is:

$$V_{k+1}(s) = R(s) + \max_a \sum_{s'} T(s, a, s')\gamma V_k(s').$$

$$V_1(s_1) = R(s_1) + \max_a \{1 \times 0.9 \times 0, \tfrac{1}{2} \times 0.9 \times 0 + \tfrac{1}{2} \times 0.9 \times 0\} = 2$$

$$\forall s_i,\ V_1(s_i) = R(s_i)$$

$$V_2(s_1) = R(s_1) + \max_a \{1 \times 0.9 \times 3, \tfrac{1}{2} \times 0.9 \times 2 + \tfrac{1}{2} \times 0.9 \times 9\} = 2 + 4.95 = 6.95$$

(b) [5 pts] Value Iteration has converged if the values of all states remain unchanged in two subsequent iterations. What is the value of state s_3 (i.e., $V^*(s_3)$) upon convergence? Describe how you get this value.

$$\underline{V_{t+1}}(s_3) = R(s_3) + \max_a \{1 \times 0.9 \times \underline{V_t(s_3)}\}$$

$$\underline{V^*}(s_3) = R(s_3) + 0.9 \cdot V^*(s_3)$$

$$0.1 V^*(s_3) = 9$$

$$V^*(s_3) = 90$$

Step number	Current state	Reward received	Action taken	Successor state
1	s_1	-10	a_1	s_1
2	s_1	-10	a_2	s_2
3	s_2	+20	a_1	s_1
4	s_1	-10	a_2	s_2

Consider a system with two states s_1 and s_2 and two actions a_1 and a_2. You performed the actions listed in table above and observed the corresponding rewards and transitions. Each step lists the current state, the reward received, the action taken, and the resulting successor state you transitioned to. For example, in Step 1, you start at state s_1, took action a_1, transitioned to state s_1 and received reward -10.

(c) [8 pts] Perform Q-learning using a learning rate $\alpha = 0.5$ and a discount factor $\gamma = 0.5$ for each step. Specifically, compute the following Q-values. You may find the Q-value update equation below helpful:

$$Q(s,a) = Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

where s is the current state, a is the action taken, s' is the successor state, and r is the reward received. Assume that all Q-values are initialized to zero.

	0	1	2	3	4
$s_1\ a_1$	0	-5	-5	-5	-5
$s_1\ a_2$	0	0	-5	-5	-5.3125
$s_2\ a_1$	0	0	0	8.75	8.75
$s_2\ a_2$	0	0	0	0	0

Q-table

- Compute $Q(s_1, a_1)$ after Step 1.

$$Q(s_1,a_1) = 0 + \frac{1}{2}\cdot\left(-10 + \frac{1}{2}\cdot 0 - 0\right) = -5$$

- Compute $Q(s_1, a_2)$ after Step 2.

$$Q(s_1,a_2) = 0 + \frac{1}{2}\cdot\left(-10 + \frac{1}{2}\cdot 0 - 0\right) = -5$$

- Compute $Q(s_2, a_1)$ after Step 3.

$$Q(s_2,a_1) = 0 + \frac{1}{2}\cdot\left(20 + \frac{1}{2}\cdot(-5) - 0\right) = 8.75$$

- Compute $Q(s_1, a_2)$ after Step 4.

$$Q(s_1,a_2) = -5 + \frac{1}{2}\cdot\left(-10 + \frac{1}{2}\cdot(8.75) + 5\right) = -5 + \frac{1}{2}\cdot(-5 + 4.375) = -5.3125$$

(d) [2 pts] What is the optimal policy π^* after these four steps? More specifically, what is the policy for each of the states below.

- $\pi^*(s_1) = a_1$
- $\pi^*(s_2) = a_1$

$$\operatorname*{argmax}_a Q^*(s,a)$$

(e) Each question is worth 1 point. Leaving a question blank is worth 0 points. **Answering a question incorrectly is worth −1 point.** This gives you an expected value of 0 for random guessing.

(i) [1 pt] [*true* or *false*] It is easier to extract optimal policies from optimal Q-values $Q^*(s,a)$ than from optimal state values $V^*(s)$.

(ii) [1 pt] [*true* or *false*] It is possible to extract an optimal policy from V-values computed via Value Iteration before it has converged.

(iii) [1 pt] [*true* or *false*] For any MDP $(S, A, T, \gamma, R, s_0)$, if we change the start state s_0, then the optimal policy is guaranteed to change as well.

(iv) [1 pt] [*true* or *false*] For any MDP $(S, A, T, \gamma, R, s_0)$, if we change the start state s_0, then the optimal policy is guaranteed to not change.

(v) [1 pt] [*true* or *false*] It is possible to extract an optimal policy from Q-values learned via Q-learning before it has converged.

(vi) [1 pt] [*true* or *false*] One disadvantage of Q-learning is that it can be used only when one does not have prior knowledge of how actions affect the environment of the agent.

(vii) [1 pt] [*true* or *false*] Q-learning can learn the optimal Q-function Q^* without ever executing the optimal policy.