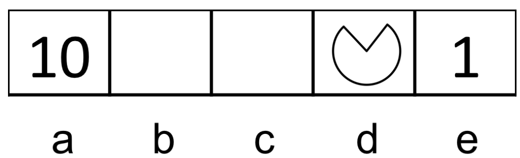
CS 4499/5599 HW 4

1 Solving MDPs

a) Consider the gridworld MDP for which *Left* and *Right* actions are 100% successful. Specifically, the available actions in each state are to move to the neighboring grid squares. From state *a*, there is also an exit action available, which results in going to the terminal state and collecting a reward of 10. Similarly, in state *e*, the reward for the exit action is 1. Exit actions are successful 100% of the time.



Let the discount factor $\gamma = 1$. Fill in the following quantities using the Bellman update rule for value iteration and keeping in mind that $V_i(s)$ is initialized to 0 for all states *s*.

$V_0(d)$ =

$V_1(d)$ =

$V_2(d)$ =

$V_3(d)$ =

$V_4(d)$ =

$V_5(d)$ =

b) For the same problem as in part a, assume that now the discount factor $\gamma = 0.2$. Fill in the following convergence values.

$V^*(a) = V_\infty(a)$ =

$V^*(b) = V_\infty(b)$ =

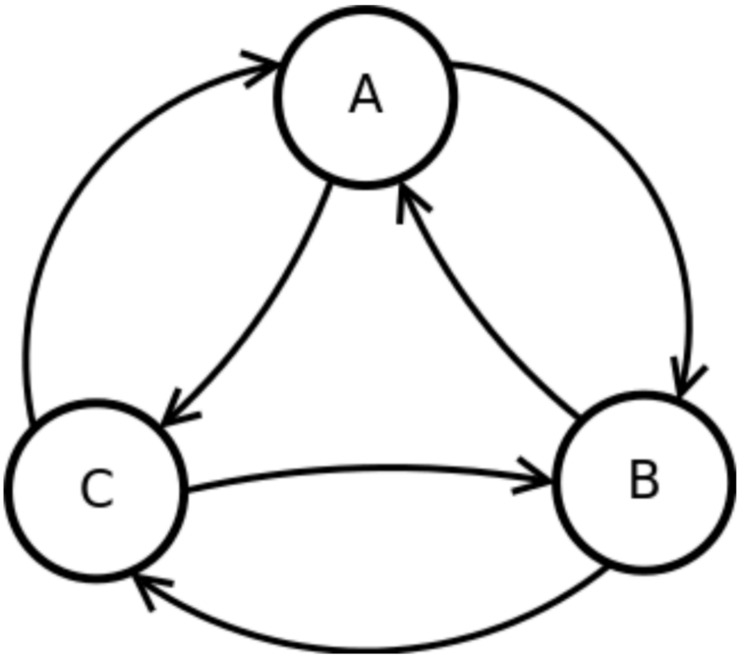
$V^*(c) = V_\infty(c)$ =

$V^*(d) = V_\infty(d)$ =

$V^*(e) = V_\infty(e)$ =

2 Value Iteration

Consider the following transition diagram, transition function and reward function for an MDP.



Discount Factor, $\gamma = 0.5$

s	a	s'	T(s,a,s')	R(s,a,s')
A	Clockwise	B	1.0	2.0
A	Counterclockwise	B	0.2	-1.0
A	Counterclockwise	C	0.8	-1.0
B	Clockwise	A	0.2	1.0
B	Clockwise	C	0.8	0.0
B	Counterclockwise	A	1.0	-1.0
C	Clockwise	A	0.8	-1.0
C	Clockwise	B	0.2	2.0
C	Counterclockwise	A	0.2	2.0
C	Counterclockwise	B	0.8	0.0

Suppose that after iteration *k* of value iteration we end up with the following values for V_k:

$V_k(A)$	$V_k(B)$	$V_k(C)$
2.100	0.560	0.680

a) What is $V_{k+1}(A)$?

Now, suppose that we ran value iteration to completion and found the following value function, V^*.

$V^*(A)$	$V^*(B)$	$V^*(C)$
2.416	0.831	0.974

b) What is $Q^*(A, \text{clockwise})$?

c) What is $Q^*(A, \text{counterclockwise})$?

d) What is the optimal action from state A?

3 Properties

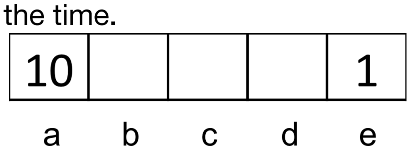
Assuming the MDP has a finite number of actions and states, and that the discount factor satisfies $0 < \gamma < 1$,

a) **True** or **False**: Value iteration is guaranteed to converge.

b) **True** or **False**: Value iteration will converge to the same vector of values (V^*) no matter what values we use to initialize V.

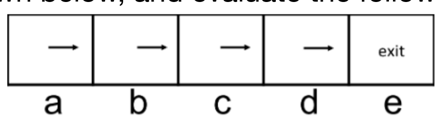
4 Policy Evaluation

Consider the gridworld MDP for which *Left* and *Right* actions are 100% successful. Specifically, the available actions in each state are to move to the neighboring grid squares. From state *a*, there is also an exit action available, which results in going to the terminal state and collecting a reward of 10. Similarly, in state *e*, the reward for the exit action is 1. Exit actions are successful 100% of the time.



Let the discount factor $\gamma = 1$.

a) Consider the policy π_1 shown below, and evaluate the following quantities for this policy.



$V^{\pi 1}(a)$ =

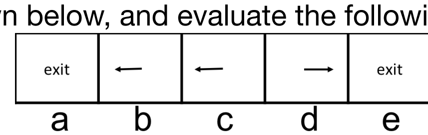
$V^{\pi 1}(b)$ =

$V^{\pi 1}(c)$ =

$V^{\pi 1}(d)$ =

$V^{\pi 1}(e)$ =

b) Consider the policy π_2 shown below, and evaluate the following quantities for this policy.



$V^{\pi 2}(a)$ =

$V^{\pi 2}(b)$ =

$V^{\pi 2}(c)$ =

$V^{\pi 2}(d)$ =

$V^{\pi 2}(e)$ =

5 Policy Iteration

Consider the gridworld MDP for which *Left* and *Right* actions are 100% successful. Specifically, the available actions in each state are to move to the neighboring grid squares. From state *a*, there is also an exit action available, which results in going to the terminal state and collecting a reward of 10. Similarly, in state *e*, the reward for the exit action is 1. Exit actions are successful 100% of the time.

10				1
a	b	c	d	e

Let the discount factor γ = 0.9. We will execute one round of policy iteration.

a) Step 1: Policy evaluation. Consider the policy π_i shown below, and evaluate the following quantities for this policy.

exit	←	→	←	exit
a	b	c	d	e

$V^{\pi_i}(a)$ =

$V^{\pi_i}(b)$ =

$V^{\pi_i}(c)$ =

$V^{\pi_i}(d)$ =

$V^{\pi_i}(e)$ =

b) Step 2: Policy improvement. Perform a policy improvement step. The current policy's values are the ones from Step 1 (so make sure you first correctly answer Step 1 before moving on to Step 2).

$\pi_{i+1}(a)$ =

$\pi_{i+1}(b)$ =

$\pi_{i+1}(c)$ =

$\pi_{i+1}(d)$ =

$\pi_{i+1}(e)$ =

6 MDPs: Pick a card

You're playing a game in which in each round the player has the option of drawing a card. In the game all cards have a value between 1 (inclusive) and 6 (inclusive). Each draw costs 1 dollar and the player **must** draw the very first round. Each time the player draws a card, the player has two possible actions:

1. *Stop*: Stop playing by collecting the dollar value of the card drawn, or
2. *Draw*: Draw again, paying another dollar

Having taken CS 4499/5599 at ISU, you decide to model this problem as an infinite horizon Markov Decision Process (MDP). The player initially starts in state *Start*, where the player only has one possible action: *Draw*. State s_i denotes the state where the drawn card has value *i.* Once a player chooses to *Stop*, the game finishes, causing the player to transition to the *End* state.

a) To solve the problem, you decide to use policy iteration. Your initial policy π is shown below. Evaluate the policy at each state, with discount γ = 1.

State	s_1	*S2*	*S3*	*S4*	*S5*	*S6*
π (s)	*Draw*	*Draw*	*Stop*	*Stop*	*Stop*	*Stop*
$V^{\pi}(s)$						

b) Having determined the values, perform a policy update to find the new policy π' . The table below shows the old policy π and has filled in parts of the updated policy π' for you. If both *Draw* and *Stop* are viable new actions for a state, write down both *Draw*/*Stop*. As previously, discount γ = 1.

State	s_1	*S2*	*S3*	*S4*	*S5*	*S6*
π (s)	*Draw*	*Draw*	*Stop*	*Stop*	*Stop*	*Stop*
π' (s)	*Draw*					*Stop*

c) Is π (s) from part (a) optimal? Justify your answer.

d) Suppose now that we are working with a discount γ ∈ [0, 1) and want to run **value iteration**. Which **one** of the following statements would hold true at convergence? If none of them are true, write the correct answer below next to “Other”.

- ○ $V^*(s_i) = \max\left\{-1 + \frac{i}{6} \ , \ \sum_j \gamma V^*(s_j)\right\}$
- ○ $V^*(s_i) = \max\left\{i \ , \ \frac{1}{6} \cdot \left[-1 + \sum_j \gamma V^*(s_j)\right]\right\}$
- ○ $V^*(s_i) = \max\left\{-\frac{1}{6} + i \ , \ \sum_j \gamma V^*(s_j)\right\}$
- ○ $V^*(s_i) = \max\left\{i \ , \ -\frac{1}{6} + \sum_j \gamma V^*(s_j)\right\}$
- ○ $V^*(s_i) = \frac{1}{6} \cdot \sum_j \max\left\{i \ , \ -1 + \gamma V^*(s_j)\right\}$
- ○ $V^*(s_i) = \frac{1}{6} \cdot \sum_j \max\left\{-1 + i \ , \ \sum_k V^*(s_j)\right\}$
- ○ $V^*(s_i) = \sum_j \max\left\{-1 + i \ , \ \frac{1}{6} \cdot \gamma V^*(s_j))\right\}$
- ○ $V^*(s_i) = \sum_j \max\left\{\frac{i}{6} \ , \ -1 + \gamma V^*(s_j)\right\}$
- ○ $V^*(s_i) = \max\left\{i \ , \ -1 + \frac{\gamma}{6} \sum_j V^*(s_j)\right\}$
- ○ $V^*(s_i) = \sum_j \max\left\{i \ , \ -\frac{1}{6} + \gamma V^*(s_j)\right\}$
- ○ $V^*(s_i) = \sum_j \max\left\{\frac{-i}{6} \ , \ -1 + \gamma V^*(s_j)\right\}$

Other: ________________________________