CS 188
Summer 2023

Discussion 5C

Q1. How do you Value It(eration)?

(a) Fill out the following True/False questions.

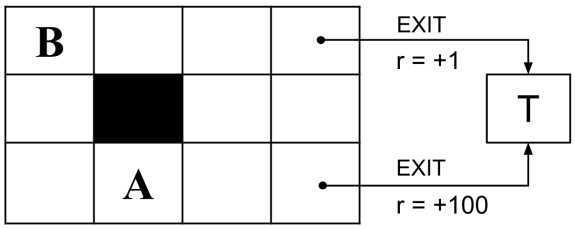
(i) ◯ True ◯ False: Let A be the set of all actions and S the set of states for some MDP. Assuming that $|A| \ll |S|$, one iteration of value iteration is generally faster than one iteration of policy iteration that solves a linear system during policy evaluation.

(ii) ◯ True ◯ False: For any MDP, changing the discount factor does not affect the optimal policy for the MDP.

The following problem will take place in various instances of a grid world MDP. Shaded cells represent walls. In all states, the agent has available actions $\uparrow, \downarrow, \leftarrow, \rightarrow$. Performing an action that would transition to an invalid state (outside the grid or into a wall) results in the agent remaining in its original state. In states with an arrow coming out, the agent has an *additional* action $EXIT$. In the event that the $EXIT$ action is taken, the agent receives the labeled reward and ends the game in the terminal state T. Unless otherwise stated, all other transitions receive no reward, and all transitions are deterministic.

For all parts of the problem, assume that value iteration begins with all states initialized to zero, i.e., $V_0(s) = 0 \ \forall s$. **Let the discount factor be $\gamma = \frac{1}{2}$ for all following parts**.

(b) Suppose that we are performing value iteration on the grid world MDP below.



(i) Fill in the optimal values for A and B in the given boxes.

$V^*(A)$: ☐ $V^*(B)$: ☐

(ii) After how many iterations k will we have $V_k(s) = V^*(s)$ for all states s? If it never occurs, write "never". Write your answer in the given box.

☐

(iii) Suppose that we wanted to re-design the reward function. For which of the following new reward functions would the optimal policy **remain unchanged**? Let $R(s, a, s')$ be the original reward function.

☐ $R_1(s, a, s') = 10R(s, a, s')$

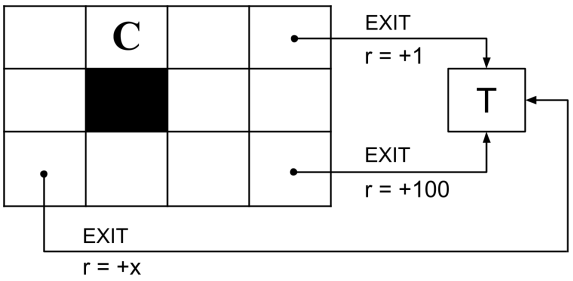
☐ $R_2(s, a, s') = 1 + R(s, a, s')$

☐ $R_3(s, a, s') = R(s, a, s')^2$

☐ $R_4(s, a, s') = -1$

☐ None

(c) For the following problem, we add a new state in which we can take the $EXIT$ action with a reward of $+x$.



(i) For what values of x is it *guaranteed* that our optimal policy π^* has $\pi^*(C) = \leftarrow$? Write ∞ and $-\infty$ if there is no upper or lower bound, respectively. Write the upper and lower bounds in each respective box.

$\square < x < \square$

(ii) For what values of x does value iteration take the **minimum** number of iterations k to converge to V^* for all states? Write ∞ and $-\infty$ if there is no upper or lower bound, respectively. Write the upper and lower bounds in each respective box.

$\square \leq x \leq \square$

(iii) Fill the box with value k, the **minimum** number of iterations until V_k has converged to V^* for all states.

$\square$

Q2. MDPs: Dice Bonanza

A casino is considering adding a new game to their collection, but need to analyze it before releasing it on their floor. They have hired you to execute the analysis. On each round of the game, the player has the option of rolling a fair 6-sided die. That is, the die lands on values 1 through 6 with equal probability. Each roll costs 1 dollar, and the player **must** roll the very first round. Each time the player rolls the die, the player has two possible actions:

1. *Stop*: Stop playing by collecting the dollar value that the die lands on, or
2. *Roll*: Roll again, paying another 1 dollar.

Having taken CS 188, you decide to model this problem using an infinite horizon Markov Decision Process (MDP). The player initially starts in state *Start*, where the player only has one possible action: *Roll*. State s_i denotes the state where the die lands on i. Once a player decides to *Stop*, the game is over, transitioning the player to the *End* state.

(a) In solving this problem, you consider using policy iteration. Your initial policy π is in the table below. Evaluate the policy at each state, with $\gamma = 1$.

State	s_1	s_2	s_3	s_4	s_5	s_6
$\pi(s)$	*Roll*	*Roll*	*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 *Roll* and *Stop* are viable new actions for a state, write down both *Roll*/*Stop*. In this part as well, we have $\gamma = 1$.

State	s_1	s_2	s_3	s_4	s_5	s_6
$\pi(s)$	*Roll*	*Roll*	*Stop*	*Stop*	*Stop*	*Stop*
$\pi'(s)$	*Roll*					*Stop*

(c) Is $\pi(s)$ from part (a) optimal? Explain why or why not.

(d) Suppose that we were now working with some $\gamma \in [0, 1)$ and wanted to run **value iteration**. Select the **one** statement that would hold true at convergence, or write the correct answer next to Other if none of the options are correct.

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

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

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

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

◯ $V^*(s_i) = \frac{1}{6} \cdot \sum_j \max\{i \ , \ -1 + \gamma V^*(s_j)\}$

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

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

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

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

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

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

◯ Other