CS 188 Summer 2021 Introduction to Artificial Intelligence Written HW 2

Due: Wednesday 07/14/2021 at 11:59pm (submit via Gradescope).

Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually

Submission: Your submission need not follow this template exactly, but you must tag where each question begins in your writeup when submitting this HW on Gradescope.

First name	
Last name	
SID	
Collaborators	

For staff use only:

Q1.	Expectimax Yahtzee	/38
Q2.	MDP: Eating Chocolate	/40
Q3.	MDPs and RL	/31
	Total	/109

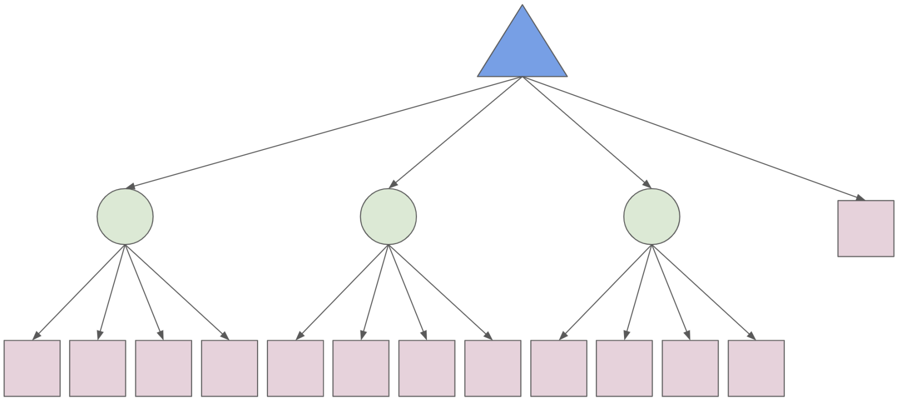
Q1. [38 pts] Expectimax Yahtzee

Consider a simplified version of the game *Yahtzee*. In this game, we have 3 dice with 4 sides each (numbered 1-4) and the game begins by rolling all 3 dice. At this point, a player can make a decision: pick one of the 3 dice to reroll, or don't reroll anything. Then, points are assigned as follows. the score is equal to the sum of all 3 dice. A reward of 10 points is given for two-of-a-kind; a reward of 15 is given to three-of-a-kind; and a reward of 7 points is given for rolling a series (1-2-3 or 2-3-4). If a special roll is achieved (series or two-of-a-kind) but the sum of dice is higher, the max is always taken.

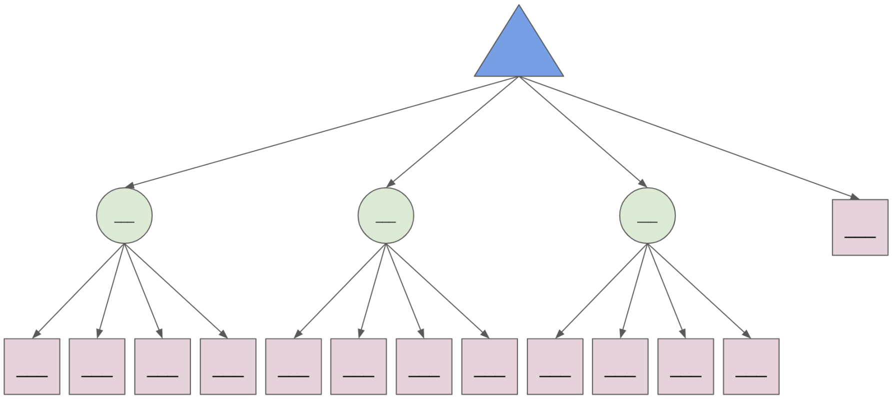
(a) Formulate this problem as an expectimax tree.

(i) [3 pts] The resulting tree for the problem is drawn below. Given a specific initial roll, the branching factor (of the player's decision) from the root node is □. The branching factor at the chance nodes is □. What do those chance nodes represent? (There are multiple solutions, you only need to write down one solution)

- Chance node 1:
- Chance node 2:
- Chance node 3:



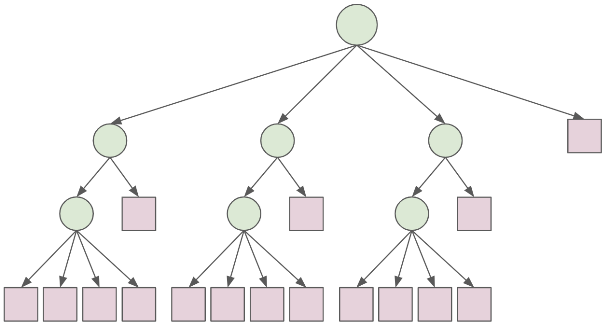
(ii) [7 pts] Given a starting roll (1,2,4) (each index in the tuple corresponding to die number 1, 2, and 3), what move should you take? Fill in the values of the expectimax tree above to justify your answer.



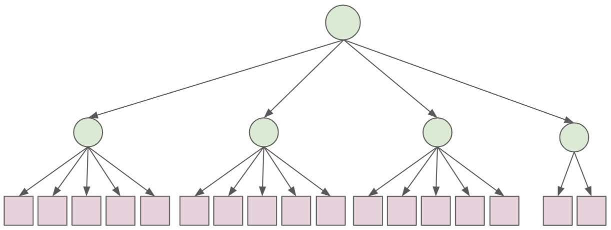
Now suppose the human player does not understand how to play the game, and as a result, they choose any action with uniform probability, regardless of the initial roll. We employ the use of a 'helpful robot' that has the power to perform the action selected by the human. Given a configuration of dice and the desired action from the human, this robot either actually implements the human's action (with probability $1 - p$) or overrides it with a 'no reroll' action (with probability $p > 0$). If the human action is already 'no reroll', then the robot does not interfere.

(b) Given a particular Yahtzee roll *roll*, Let A, B, C and D be the expected reward of performing actions 'reroll die 1', 'reroll die 2', 'reroll die 3', and 'no reroll', respectively.

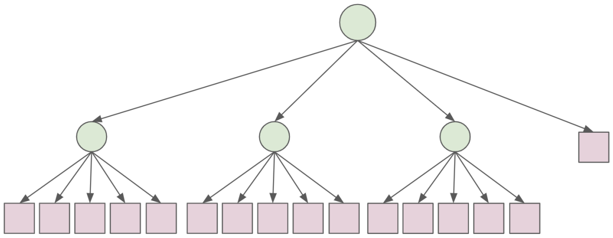
(i) [3 pts] Which of the following trees best represent the expectimax tree, after accounting for the presence of the 'helpful robot'. Please use chance nodes to denote all the non-deterministic action results.



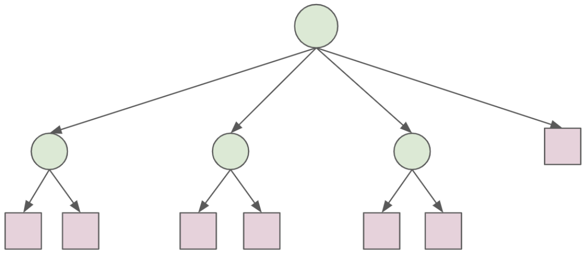
○



○



○



○

○ None of the above.

(ii) [2 pts] In terms of A, B, C and D, what is R_H, the expected reward after the human takes an action without the robot intervening? What is R_{RH}, the expected reward after the robot intervenes and performs its chosen action? Please **show all steps of your work** and **write your expression into the form of** $X + Yp$, where X and Y are expressions that contain A, B, C and D but not p.

$R_H =$

$R_{RH} =$

(iii) [1 pt] What is the condition for our "helpful robot" to strictly increase expected reward? Write the condition above using only $A, B, C, D, >$.

(iv) [2 pts] In one sentence, please describe the situation when the condition above is true.

(c) Your friend Diana argues that a helpful robot should not only override the human player's "reroll" choice with probability p (and replace it with a "no reroll"), but also override the human player's "no reroll" choice with probability p (and replace it with the outcome of selecting one of the 3 dice at random and rerolling that dice). She would like to implement the "Dianabot", but would like your help with drawing the new expectimax tree

(i) [10 pts] Draw the expectimax tree for "Dianabot". You need to draw out the tree with all nodes in their correct shapes; you do not need to label any values in the tree. Hint: you can start by modifying the expectimax tree for the 'helpful robot'.

(ii) [5 pts] What is the expected reward for a random player under "Dianabot"'s assistance? Again, please **show all steps of your work** and **write your expression into the form of** $X + Yp$, where X and Y are expressions that contain A, B, C and D but not p.

$R_{DH} =$

(iii) [3 pts] Write the condition for the "Dianabot" to strictly increase expected reward above using only $A, B, C, D, >$.

(iv) [2 pts] In one sentence, please describe the situation when the condition above is true.

Q2. [40 pts] MDP: Eating Chocolate

We have a chocolate bar of dimensions 1×8, which contains 8 squares. Most of these squares are delicious chocolate squares, but some of them are poison! Although the chocolate and poison squares are visually indistinguishable, someone has told us which ones are which. The layout for our chocolate bar is shown below with P indicating poison squares.

P				P		P	

Eating a chocolate square immediately gives a reward of 1 and eating a poison square immediately gives a reward of -2. **Starting from the right end of the bar**, there are 3 possible actions that we can take (at each step), and all of these actions cause non-deterministic transitions as follows:

- Action b_1: Try to bite 1 square, which will result in you actually eating 0, 1, or 2 squares with equal probability.
- Action b_2: Try to bite 2 squares, which will result in you actually eating 1, 2, or 3 squares with equal probability.
- Action *Stop*: Stop biting, which will end the game definitively and result in no reward.

(a) [10 pts] Formulate this problem as an MDP. Fill in the blanks.

States: □ through □ indicating the length of the remaining bar, Done state

Actions: □

Transitions:

$$T(s, b_1, s') = \boxed{}, s > 2, s' \in \{s, s-1, s-2\}$$

$$T(2, b_1, s') = \begin{cases} \boxed{} & \text{if } s' = 2 \\ \boxed{} & \text{if } s' = 1 \\ \boxed{} & \text{if } s' = \textit{Done} \end{cases}$$

$$T(1, b_1, s') = \begin{cases} \boxed{} & \text{if } s' = 1 \\ \boxed{} & \text{if } s' = \textit{Done} \end{cases}$$

$$T(s, b_2, s') = \boxed{}, s > 3, s' \in \boxed{}$$

$$T(3, b_2, s') = \boxed{}, s' \in \boxed{}$$

$$T(2, b_2, s') = \begin{cases} \boxed{} & \text{if } s' = \boxed{} \\ \boxed{} & \text{if } s' = \boxed{} \end{cases}$$

$$T(1, b_2, \textit{Done}) = 1$$

$$T(s, \textit{Stop}, \textit{Done}) = 1$$

$$T(s, a, s') = 0 \text{ otherwise}$$

Rewards: Skip the explicit formulation of the rewards for this problem.

(b) Value Iteration

(i) [5 pts] Perform value iteration for 3 iterations, using $\gamma = 1$. What are the values for each state, after each of these iterations? Please fill in all the the empty boxes in the chart below. You do not need to compute values for boxes marked with '-' in $V_3(s)$.

State	1	2	3	4	5	6	7	8
$V_0(s)$	0	0	0	0	0	0	0	0
$V_1(s)$								
$V_2(s)$								
$V_3(s)$	-	-	-					

(ii) [1 pt] We consider value iteration to have converged by iteration k if $\forall s, V_{k+1}(s) = V_k(s)$. Did the values converge by iteration 2?

○ Yes ○ No

(iii) [1 pt] Given that we know a $V(s)$ for all states, how do we extract a policy $\pi(s)$ from that value function? Your answer should be a one-line math expression which uses some or all of the following: $s, a, s', V, R, T, \gamma, \sum$, argmax, max.

$\pi(s) =$ □

(iv) [3 pts] What is the resulting policy after extracting it from the values $V_2(s)$? If applicable, write all of the valid actions for a given state.

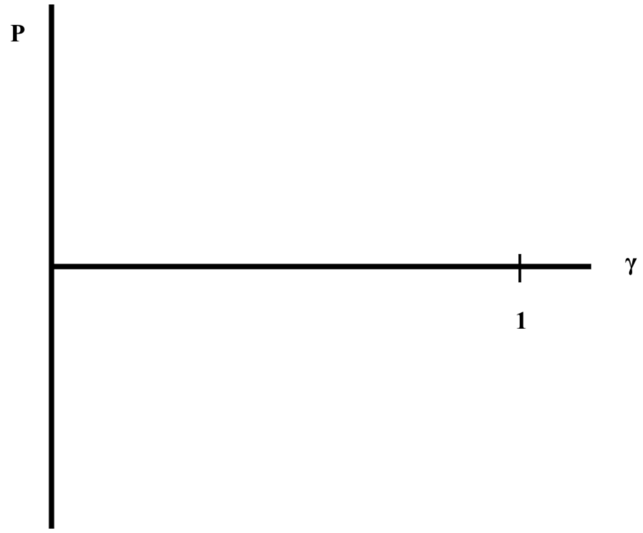
State	1	2	3	4	5	6	7	8
$\pi_3(s)$								

For the rest of the problem, assume we have deterministic transitions. This means that taking the b_1 action bites exactly 1 square and taking the b_2 actions bites exactly 2 squares. Note that if you take b_1, then your reward for that step is the reward associated with that one square, and if you take b_2, then your reward for that step is the sum of rewards from the two squares.

(c) **(i)** [6 pts] Let's say that you took a sequence of 3 single bites, so you are now in state $s = 5$. At this point, you can either stop or continue to bite (taking actions b_1 or b_2). For what range of discount values γ would you choose to stop versus continue to bite? Show your work and explain your answer.

(ii) [4 pts] For this part, assume the same situation as the previous part, where you took 3 single bites already and you are now in state $s = 5$. Think about how the behavior of the optimal policy at this point changes, as we vary (a) the reward associated with eating a poison square and (b) the discount γ from 0 to 1. First, what's the mathematical condition that must hold for us to choose to continue to bite:

Plot this condition in the plot below by shading in the areas where we will continue to bite. Label important points, including the point where $P = -2$ with the threshold γ value you found in part (c.i).



(d) **(i)** [4 pts] Define/create a policy below such that performing policy evaluation on that policy would give these values indicated here. Use $\gamma = 1$

State	1	2	3	4	5	6	7	8
$V^{\pi}(s)$	0	-1	2	3	0	2	-1	0
$\pi(s)$								

(ii) [6 pts] Perform one step of policy improvement on the previous policy you found in part (d.i). Please show your work.

State	1	2	3	4	5	6	7	8
$\pi_{i+1}(s)$								

Was the policy in part (d.i) optimal: ○ Yes ○ No

Q3. [31 pts] MDPs and RL

The agent is in a 2×4 gridworld as shown in the figure. We start from square 1 and finish in square 8. When square 8 is reached, we receive a reward of $+10$ at the game end. For anything else, we receive a constant reward of -1 (you can think of this as a time penalty).

1	2	3	4
5	6	7	8

The actions in this MDP include: up, down, left and right. The agent cannot take actions that take them off the board. In the table below, we provide initial non-zero estimates of Q values (Q values for invalid actions are left as blanks):

Table 1

	action=up	action=down	action=left	action=right
state=1		Q(1, down)=4		Q(1, right)=3
state=2		Q(2, down)=6	Q(2, left)=4	Q(2, right)=5
state=3		Q(3, down)=8	Q(3, left)=5	Q(3, right)=7
state=4		Q(4, down)=9	Q(4, left)=6	
state=5	Q(5, up)=5			Q(5, right)=6
state=6	Q(6, up)=4		Q(6, left)=5	Q(6, right)=7
state=7	Q(7, up)=6		Q(7, left)=6	Q(7, right)=8

(a) Your friend Adam guesses that the actions in this MDP are fully deterministic (e.g. taking down from 2 will land you in 6 with probability 1 and everywhere else with probability 0). Since we have full knowledge of T and R, we can thus use the Bellman equation to improve (i.e., further update) the initial Q estimates.

Adam tells you to use the following update rule for Q values, where he assumes that your policy is greedy and thus does $\max_a Q(s, a)$. The update rule he prescribes is as follows:

$$Q_{k+1}(s, a) = \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma \max_{a'} Q_k(s', a')]$$

(i) [1 pt] Perform one update of $Q(3, \text{left})$ using the equation above, where $\gamma = 0.8$. You may break ties in any way.

(ii) [1 pt] Perform one update of $Q(3, \text{down})$ using the equation above, where $\gamma = 0.8$.

(iii) [3 pts] For the Q update rule prescribed above, how is it different from the Q learning update that we saw in lecture, which is $Q_{k+1}(s, a) = (1 - \alpha)Q_k(s, a) + \alpha * \text{sample}$?

(b) After observing the agent for a while, Adam realized that his assumption of T being deterministic is wrong in one specific way: **when the agent tries to legally move down, it occasionally ends up moving left instead** (except from grid 1 where moving left results in out-of-bound). All other movements are still deterministic.

Suppose we have run the Q updates outlined in the equation above until convergence, to get $Q^*_{wrong}(s, a)$ under the original assumption of the wrong (deterministic) T. Suppose $Q^*_{correct}(s, a)$ denotes the Q values under the new correct T. Note that you don't explicitly know the exact probabilities associated with this new T, but you know that it qualitatively differs in the way described above. As prompted below, list the set of (s, a) pairs where $Q^*_{wrong}(s, a)$ is either an over-estimate or under-estimate of $Q^*_{correct}(s, a)$.

(i) [3 pts] List of (s, a) where $Q^*_{wrong}(s, a)$ is an over-estimate. Explain why.

(ii) [3 pts] List of (s, a) where $Q^*_{wrong}(s, a)$ is an under-estimate (and why):

(c) [2 pts] Suppose that we have a mysterious oracle that can give us either all the correct Q-values $Q(s, a)$ or all the correct state values $V(s)$. Which one do you prefer to be given if you want to use it to find the optimal policy, and why?

(d) [2 pts] Suppose that you perform actions in this grid and observe the following episode: 3, right, 4, down, 8 (terminal).

With learning rate $\alpha = 0.2$, discount $\gamma = 0.8$, perform an update of $Q(3, right)$ and $Q(4, down)$. Note that here, we update Q values based on the sampled actions as in TD learning, rather than the greedy actions.

(e) [2 pts] One way to encourage an agent to perform more exploration in the world is known as the "ϵ-greedy" algorithm. For any given policy $\pi(s)$, this algorithm says to take the original action $a = \pi(s)$ with probability $(1 - \epsilon)$, and to take a random action (drawn from a uniform distribution over all legal actions) with probability ϵ. If ϵ can be tuned, would you assign it to be a high or low value at the beginning of training? What about at the end of the training? Please answer both questions and justify your choices.

(f) Instead of using the "ϵ-greedy" algorithm, we will now do some interesting exploration with softmax. We first introduce a new type of policy: A stochastic policy $\pi(a|s)$ represents the probability of action a being prescribed, conditioned on the current state. In other words, the policy is a now a distribution over possible actions, rather than a function that outputs a deterministic action.

Let's define a new policy as follows:

$$\pi(a|s) = \frac{e^{Q(s,a)}}{\sum_{a'} e^{Q(s,a')}}$$

(i) [2 pts] Suppose we are at square 3 in the grid and we want to use the originally provided Q values from the table. What is the probability that this policy will tell us to go right? What is the probability that this policy will tell us to go left? Note that the sum over actions prescribed above refers to a sum over legal actions.

(ii) [2 pts] How is this exploration strategy qualitatively different from "ϵ-greedy"?

(g) [10 pts] Your friend Cody argues that we could still explicitly calculate Q updates (like Adam's approach in part (a)) even if we don't know the true underlying transition function $T(s, a, s')$, because he believes that our T can be roughly approximated from samples.

(i) [3 pts] Suppose you collect 1,000 transitions from $s = 3, a = \mathit{Down}$, in the form of (s_{start}, a, s_{end}). Describe how you can use these samples to compute $T_{approx}(s = 3, a = \mathit{Down}, s')$, which is an approximation of the true underlying (unknown) $T(s, a, s')$.

(s =3, a = Down, s'= 6)	(s = 3, a= Down, s'=7)
99	901

(ii) [2 pts] Now perform one step of q-value iteration based on your transition model computed above.