UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

CS440/ECE448 Artificial Intelligence

Practice Exam 2

Spring 2022

Exam 3 will be May 13, 2022

Your Name: ______________________________

Your NetID: ______________________________

Instructions

- Please write your name and NetID on the top of every page.
- This will be a CLOSED BOOK exam. You will be permitted to bring tweo 8.5x11 pages of handwritten notes (front & back).
- Calculators are not permitted. You need not simplify explicit numerical expressions.
- The actual exam will be about 1/6 material from exam 1, about 1/6 material from exam 2, and about 2/3 material from the last third of the course. This practice exam contains only material from the last third of the course.

Question 1 ***(0 points)***

Consider the following game:

	Player A: Action 1	Player A: Action 2
Player B: Action 1	A=3 B=2	A=0 B=0
Player B: Action 2	A=1 B=1	A=2 B=3

(a) Find dominant strategies (if any).

Solution: A dominant strategy is defined as a strategy whose outcome is better for the player regardless of the strategy chosen by the other player. Let's first look for dominant strategies for A: Suppose B chooses Action1. A gets 3 if it chooses Action1 or 0 if it chooses Action2. So it shoould choose Action1. Now suppose B chooses Action2. A gets 1 if it chooses Action1 or 2 if it chooses Action2. So it should choose Action2. Thus there is no dominant strategy for A. Let's look at B: Suppose A chooses Action1. B gets 2 if it chooses Action1 or 1 if it chooses Action2. So it should choose Action1. Now suppose A chooses Action2. B gets 0 if it chooses Action1 or 3 if it chooses Action2. So it should choose Action2. Thus there is also no dominant strategy for B.

(b) Find pure strategy equilibria (if any).

Solution: A Nash Equilibrium is a set of strategies such that no player can get a bigger payoff by switching strageties, provided the other player sticks with the same strategy. There are two: (A: Action1, B: Action1) or (A: Action2, B: Action2).

Question 2 ***(0 points)***

In each square, the first number refers to payoff for the player whose moves are shown on the row-label, the second number refers to payoff for the player shown on the column label.

	A’	B’	C’
A	0, 0	25, 40	5, 10
B	40, 25	0, 0	5, 15
C	10, 5	15, 5	10, 10

(a) Are there any dominant strategies? If so, what are they? If not, why not?

Solution: No. The best move, for each player, depends on what the other player does.

(b) Are there any pure-strategy Nash equilibria? If so, what are they? If not, why not?

Solution: (A,B), (B,A), (C,C)

(c) Are there any Pareto-optimal solutions? If so, what are they? If not, why not?

Solution: (A,B) and (B,A).

Question 3 ***(0 points)***

Suppose that both Alice and Bob want to go from one place to another. There are two routes R1 and R2. The utility of a route is inversely proportional to the number of cars on the road. For instance, if both Alice and Bob choose route R1, the utility of R1 for each of them is 1/2.

(a) Write out the payoff matrix.

Solution:

	Alice R1	Alice R2
Bob R1	A:0.5, B:0.5	A:1, B:1
Bob R2	A:1, B:1	A:0.5, B:0.5

(b) Is this a zero-sum game? Why or why not?

Solution: No. The rewards for Bob and Alice do not sum to zero.

(c) Find dominant strategies, if any. If there are no dominant strategies, explain why not.

Solution: There is no dominant strategy for either player. The best strategy for each player depends on the strategy of the other player.

(d) Find pure strategy equilibria, if any. If there are no pure strategy equilibria, explain why not.

Solution: There are two: (Alice=R1,Bob=R2) and (Alice=R2,Bob=R1).

(e) Find the mixed strategy equilibrium.

Solution: Alice chooses R1 with probability p, and R2 with probability $1-p$. p must be chosen so that Bob's reward is independent of the action he takes.

- Bob's Reward(R1)$= 0.5p+(1-p) = 1-0.5p$
- Bob's Reward(R2)$= p+0.5(1-p) = 0.5+0.5p$

Setting the two rewards equal, we find $p = 0.5$.

Question 4 ***(0 points)***

The "Battle of the Species" game is defined as follows. Imagine a cat and a dog have agreed to meet for the evening, but they forgot whether they were going to meet at a frisbee field or an aquarium. The dog prefers the frisbee field and the cat prefers the aquarium. The payoff for each one's preferred activity is 4 and the payoff for the non-preferred activity is 3 – assuming the cat and the dog end up at the same place. If they end up at different places, each gets a 1 if they are at their preferred place, and 0 if they are at their non-preferred place.

(a) Give the normal form (matrix) representation of the game.

Solution:

	Dog: Frisbee	Dog: Aquarium
Cat: Frisbee	C:3,D:4	D:0,C:0
Cat: Aquarium	C:1,D:1	C:4,D:3

(b) Find dominant strategies (if any). Briefly explain your answer.

Solution: None. A dominant strategy is a strategy that maximizes the player's payoff regardless of what the other player does. In this case, if one player chooses frisbee, the other one should choose frisbee, and if one chooses aquarium, the other one should choose aquarium. Therefore, there is no dominant strategy.

(c) Find pure strategy equilibria (if any). Briefly explain your answer.

Solution: (Dog: frisbee; Cat: frisbee); (Dog: aquarium; Cat: aquarium). From either of these two states, no player can get a bigger payoff from changing actions unilaterally.

Question 5 *(0 points)*

Give an example of a coordination game and an anti-coordination game. For each game, write down its payoff matrix, list dominant strategies and pure strategy Nash equilibria (if any).

Solution: The stag hunt is a coordination game. The payoff matrix (player 1's payoff listed first inside each ssquare) is:

	Player 2: Cooperate	Player 2: Defect
Player 1: Cooperate	2, 2	0,1
Player 1: Defect	1,0	1,1

It has no dominant strategy. It has two pure-strategy Nash equilibria: (C,C) and (D,D).

The game of Chicken is an anti-coordination game. The payoff matrix (player 1's payoff listed first inside each ssquare) is:

	Player 2: Chicken	Player 2: Drive
Player 1: Chicken	0, 0	-1,1
Player 1: Drive	1,-1	-10,-10

It has no dominant strategy. It has two pure-strategy Nash equilibria: (C,D) and (D,C).

Question 6 *(0 points)*

In the lectures, we covered dominant strategies of simultaneous move games. We can also consider minimax strategies for such games, defined in the same way as for multi-player alternating games, except that now, both players make their decision before they have seen what the other player will do. What would be the minimax strategies in the Prisoner's Dilemma, Stag Hunt, and Game of Chicken? If both players follow the minimax strategy, does the game outcome differ from the Nash equilibria? When/why would one prefer to choose a minimax strategy rather than a Nash equilibrium?

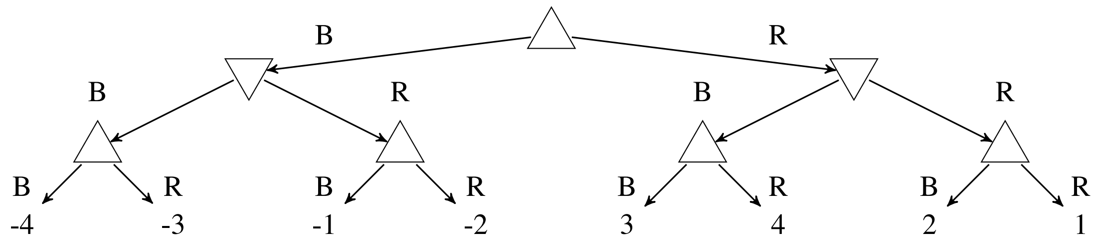
Solution: Minimax solution maximizes, over all of your possible actions, the minimum, over all of your opponent's possible actions, of your reward.

- Prisoner's Dilemma: Defect. Result is also the Nash equilibrium.
- Stag Hunt: take the Hare. Result is one of the two Nash equilibria.
- Game of Chicken: chicken out. Result is not a Nash equilibrium.

Nash equilibrium is the outcome achieved if each player, knowing the other player's action, has no reason to change their own action: it assumes that you know the other player's action. Minimax makes more sense if you want to limit your losses, and have no way to predict the other player's behavior.

Question 7 ***(8 points)***

The following **minimax tree** shows all possible outcomes of the RED-BLUE game. In this game, Max plays first, then Min, then Max. Each player, when it's their turn, chooses either a blue stone (B) or a red stone (R); after three turns, Max wins the number of points shown (negative scores indicate a win for Min).



(a) (3 points) Max could be a Reflex Agent, following a set of predefined IF-THEN rules, and could still play optimally against Min, even if Min is not rational. To do so, Max needs just three rules of the form "If the stones already chosen are ____, then choose a ____ stone." Write those three rules in that form.

> **Solution:**
>
> - If no stones have been chosen, then choose a Red stone.
> - If the stones already chosen are (R,B), then choose a Red stone.
> - If the stones already chosen are (R,R), then choose a Blue stone.

(b) (2 points) Recall that an $\alpha - \beta$ search prunes the largest possible number of moves if there is extra information available to the players that permits them to evaluate the moves in the best possible order. IN GENERAL (not just for this game tree),

- In what order should the moves available to MAX be evaluated, in order to prune as many moves as possible?
- In what order should the moves available to MIN be evaluated, in order to prune as many moves as possible?

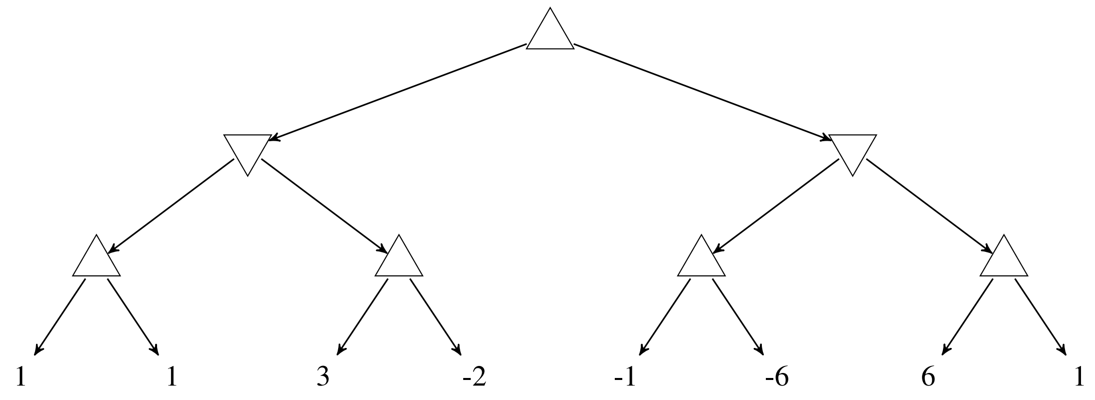
> **Solution:**
>
> - Moves available to MAX should be evaluated in order of descending value, starting with the highest-value move.
> - Moves available to MIN should be evaluated in order of ascending value, starting with the lowest-value move.

(c) (3 points) Re-draw the minimax tree for the RED-BLUE game so that, if moves are always evaluated from left to right, the $\alpha - \beta$ search only needs to evaluate 5 of the 8 terminal states.

> **Solution:** The first two leaves should be (1,2) or (2,1). The next two leaves should be (3,4) or (4,3). The last four leaves should be (-1,2,-3,-4), in any order.

Question 8 ***(0 points)***

Two players, MAX and MIN, are playing a game. The game tree is shown below. Upward-pointing triangles denote decisions by MAX; downward-pointing triangles denote decisions by MIN. Numbers on the terminal nodes show the final score: MAX seeks to maximize the final score, MIN seeks to minimize the final score.



(a) Write the minimax value of each nonterminal node (each upward-pointing or downward-pointing triangle) next to it.

Solution: From top to bottom, left to right, the values are 1, 1, -1, 1, 3, -1, 6.

(b) Suppose that the minimax values of the nodes at each level are computed in order, from left to right. Draw an X through any edge that would be pruned (eliminated from consideration) using alpha-beta pruning.

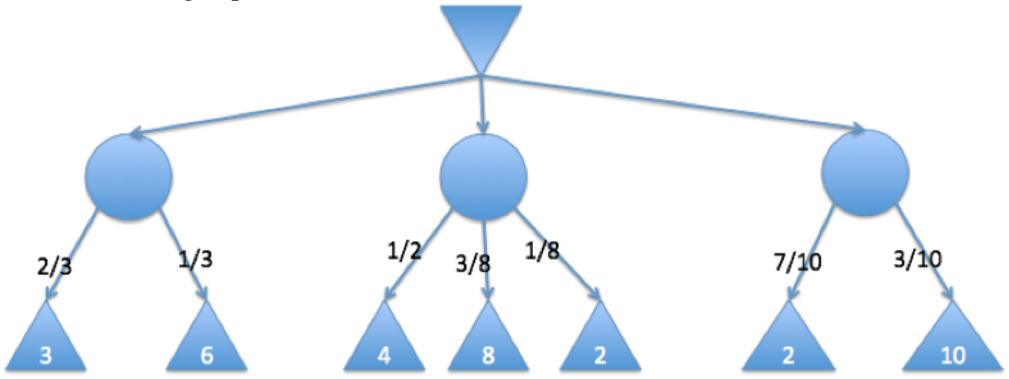
Solution: The 4th edge at the bottom level, and the 4th edge at the middle level, would both be pruned.

(c) In this game, alpha-beta pruning did not change the minimax value of the start node. Is there any deterministic two-player game tree in which alpha-beta pruning changes the minimax value of the start node? Why or why not?

Solution: No. Alpha-beta pruning only prunes branches that have no effect on the start node.

Question 9 ***(0 points)***

Consider the following expectiminimax tree:



Circle nodes are chance nodes, the top node is a min node, and the bottom nodes are max nodes.

(a) For each circle, calculate the node values, as per expectiminimax definition.

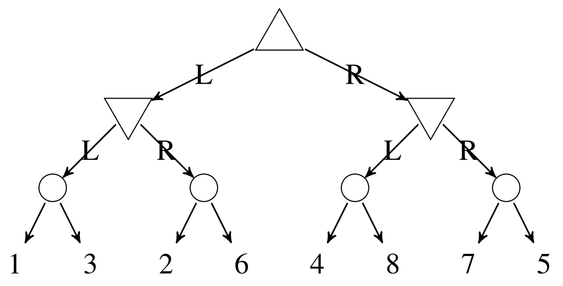
Solution: From left to right: 4, 5.25, 4.4.

(b) Which action should the min player take?

Solution: The first action.

Question 10 ***(5 points)***

Consider a game with eight cards ($c \in \{1,2,3,4,5,6,7,8\}$), sorted onto the table in four stacks of two cards each. MAX and MIN each know the contents of each stack, but they don't know which card is on top. The game proceeds as follows. First, MAX chooses either the left or the right pair of stacks. Second, MIN chooses either the left or the right stack, within the pair that MAX chose. Finally, the top card is revealed. MAX receives the face value of the card (c), and MIN receives $9-c$. The resulting expectiminimax tree is as follows:



(a) (2 points) Assume that the two cards in each stack are equally likely. What is the value of the top MAX node?

Solution: Propagating backward using expectiminimax, we find that the value of the top node is 6.

(b) (3 points) Consider the following rule change: after MAX chooses a pair of stacks, he is permitted to look at the top card in any one stack. He must show the card to MIN, then replace it, so that it remains the top card in that stack. Define the belief state, b, to be the set of all possible outcomes of the game, i.e., the starting belief state is the set $b = \{1,2,3,4,5,6,7,8\}$; the PREDICT operation modifies the belief state based on the action of a player, and the OBSERVE operation modifies the belief state based on MAX's observation. Suppose MAX chooses the action R. He then turns up the top card in the rightmost deck, revealing it to be a 7. What is the resulting belief state?

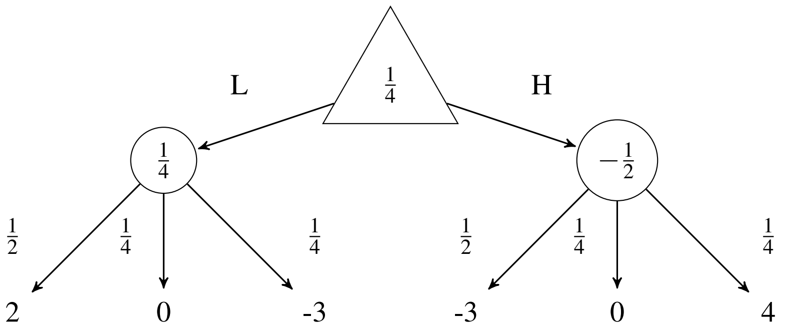
Solution: After MAX chooses the right set of stacks, the PREDICT update step results in a belief state of $b = \{4,8,7,5\}$. After he looks at the rightmost deck and finds that it contains a 7, the OBSERVE update step restricts the belief state to $b = \{4,8,7\}$.

Question 11 ***(0 points)***

Consider the following game, called "High/Low." There is an infinite deck of cards, half of which are 2's, one quarter are 3's, and one quarter are 4's. The game starts with a 3 showing. After each card, you say "High" or "Low," and a new card is flipped. If you are correct (e.g., you say "High" and then the next card is higher than the one showing), you win the points shown on the new card. If there is a tie (the next card equals the one showing), you get zero points. If you are wrong (e.g., you say "High" and then the next card is lower than the one showing), then you lose the amount of the card that was already showing.

Draw the expectimax tree for the first round of this game and write down the expected utility of every node. What is the optimal policy assuming the game only lasts one round?

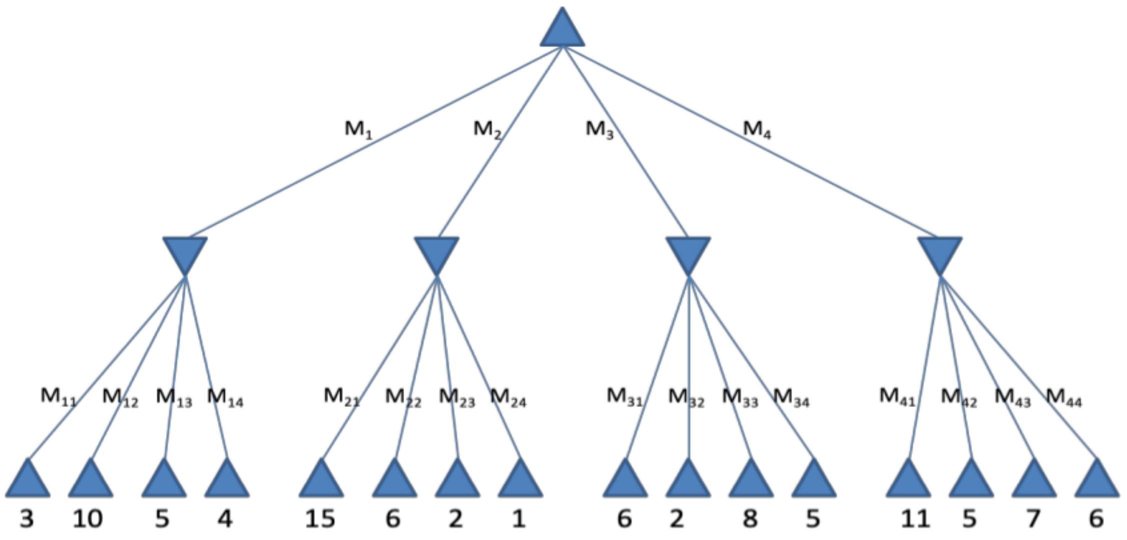
Solution:



Optimal policy: L

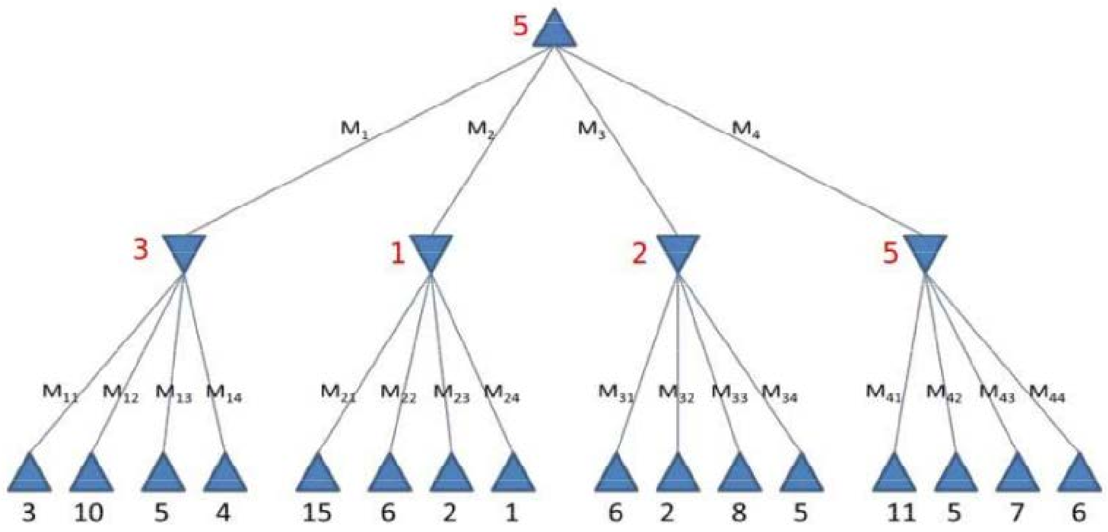
Question 12 ***(0 points)***

Consider the following game tree (MAX moves first):



(a) Write down the minimax value of every non-terminal node next to that node.

Solution:

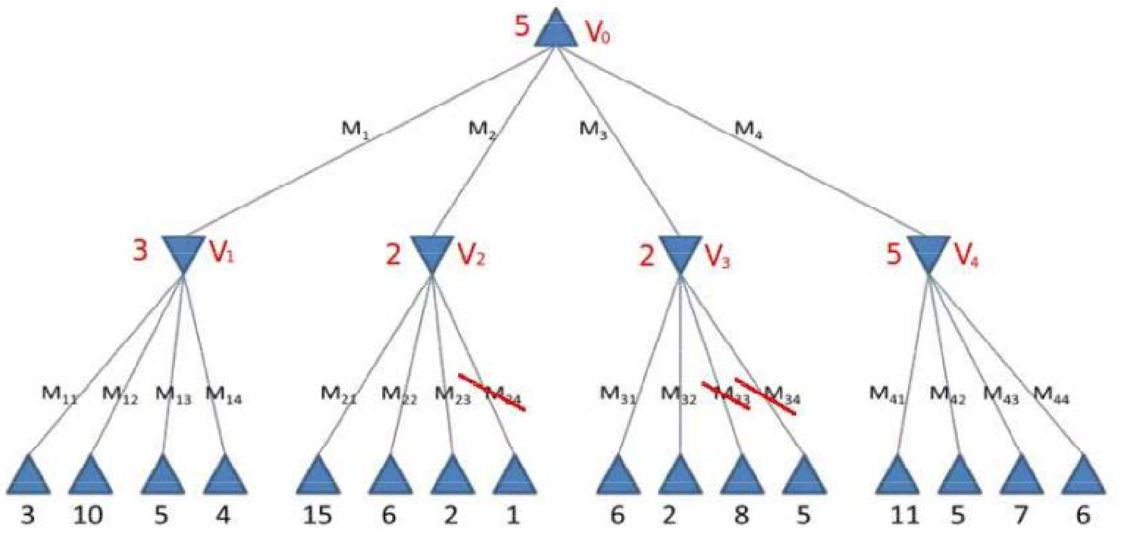


(b) How will the game proceed, assuming both players play optimally?

Solution: The game will choose the max value on depth 1, taking route M_4. It will then take the minimum value on depth 2 that is child of the chosen node, and hence take M_{42}.

(c) Cross out the branches that do not need to be examined by alpha-beta search in order to find the minimax value of the top node, assuming that moves are considered in the non-optimal order shown.

Solution:



(d) Suppose that a heuristic was available that could re-order the moves of both max (M_1, M_2, M_3, M_4) and min ($M_{11}, \ldots, M_{44}$) in order to force the alpha-beta algorithm to prune as many nodes as possible. Which max move would be considered first: M_1, M_2, M_3, or M_4? Which of the min moves ($M_{11}, \ldots, M_{44}$) would have to be considered?

Solution: The first max move to be considered would be V_4, because it allows us to set the highest α. Only 7 of the min moves would be considered: M_{41} through M_{44} would have to be considered to determine that $\alpha = 5$, and then (if the heuristic magically sorts moves in order for us), we would consider M_{32}, M_{24}, and M_{11}, find that all of them have values below α, and prune away their parents.

Question 13 ***(0 points)***

What are the main challenges of adversarial search as contrasted with single-agent search? What are some algorithmic similarities and differences?

Solution: The biggest difference is that we are unaware of how the opponent(s) will act. Because of this our search cannot simply consider my own moves, it must also figure out how my opponent will act at each level, thus effectively doubling the number of levels over which I have to search. Since the number of levels is the exponent in the computational complexity, this makes computational complexity much harder.

Question 14 ***(0 points)***

What additional difficulties does dice throwing or other sources of uncertainty introduce into a game?

Solution: Uncertainties introduce probabilities into the game. Expectiminimax is used to find solutions for these type of games. Expectiminimax doubles the number of levels as compared to minimax, because there is randomness after every play. Expectiminimax has nasty branching factor and often times defining evaluation functions and pruning algorithms are difficult.

Question 15 ***(0 points)***

How can randomness be incorporated into a game tree? How about partial observability (imperfect information)?

Solution: Randomness is incorporated using the expectiminimax algorithm, in which max tries to maximize the expected score, min tries to minimize the expected score. Partial observability is incorporated using a minimax state tree in which neither player knows for sure which state they're in; the max player chooses an action that maximizes the minimum payoff over all of the states he might be in.

Question 16 *(7 points)*

ATARA is an Automatic Telephone-based Airplane Reservation Agent.

In order to make an airplane reservation, ATARA needs to learn the user's starting city, ending city, and date of travel (she always asks in that order). When she starts each dialog, she knows none of these things.

During each of her dialog turns, ATARA has the option of asking for 1 or 2 pieces of information. Unfortunately, her speech recognizer makes mistakes. If she asks for 1 piece of information, she always gets it. If she asks for 2 pieces of information, then she gets both pieces of information with probability $\left(\frac{1}{2}\right)$, but with probability $\left(\frac{1}{2}\right)$, she gets nothing.

ATARA receives a reward of $R(s) = 10$, and ends the dialog, when she has correctly recognized all 3 pieces of information. Otherwise, she gets a reward of $R(s) = -1$ for each dialog turn during which she has not finished the dialog.

(a) (1 point) What is the set of states for this Markov decision process?

Solution: The states are $s \in \{0, 1, 2, 3\}$, specifying the number of pieces of information ATARA has collected.

(b) (1 point) What is the set of actions?

Solution: The actions are $a \in \{1, 2\}$, representing the number of pieces of information that ATARA requests.

(c) (3 points) Write the transition probability table $P(s'|s,a)$.

Solution:

S'

(s,a)	0	1	2	3
$(0,1)$	0	1	0	0
$(0,2)$	1/2	0	1/2	0
$(1,1)$	0	0	1	0
$(1,2)$	0	1/2	0	1/2
$(2,1)$	0	0	0	1
$(2,2)$	0	0	1/2	1/2

(d) (2 points) Use value iteration to find $U(s)$, the utility of each state, assuming a discount factor of $\gamma = 1$.

Solution: The utility estimate at each iteration is given as follows; after $t = 5$, the utility no longer changes.

t	0	1	2	3
1	0	0	0	0
2	-1	-1	-1	10
3	-2	3.5	9	10
4	2.5	8	9	10
5	7	8	9	10

$R(0) = R(1) = R(2) = -1 \quad R(3) = 10$

$-1 + \gamma \max\left(1 \cdot (-1), \frac{1}{2} \cdot (-1) + \frac{1}{2}(-1)\right) = -2$

$-1 + \gamma \max\left(1 \cdot (-1), \frac{1}{2}(-1) + \frac{1}{2}(10)\right) = 3.5$

Question 17 ***(0 points)***

After t iterations of the "Value Iteration" algorithm, the estimated utility $U(s)$ is a summation including terms $R(s')$ for the set of states s' that can be reached from state s in at most $t-1$ steps.

[✓ **True**]

◯ False

Explain:

Solution: Value iteration starts with $U(s) = 0$. Each iteration updates $U(s)$ by adding $R(s)$, plus the maximum over all actions of the expected utility $U(s')$ of the state s' that can be reached from state s in one step. In t iterations of this algorithm, one accumulates rewards from states that are up to $t-1$ steps away.

	Simplified GridWorld Column Number 1	2
1	-0.04	-0.04
2	-1.00	1.00
3	0.00	-0.04

Question 18 ***(0 points)***

Consider a simplified version of GridWorld, shown above. The grid above shows the reward, $R(s)$, associated with each state. The robot starts in the state with $R(s) = 0.00$; if it reaches either the state with $R(s) = 1.00$ or $R(s) = -1.00$, the game ends.

The transition probabilities are simpler than the ones used in lecture. Let the action variable, a, denote the state to which the robot is trying to move. If the robot tries to move out of the maze, it always stays in the state where it started. If the robot tries to move to any state that is a neighbor of the state it currently occupies, then it either succeeds (with probability 0.8), or else it remains in the same state (with probability 0.2). To put the same transition probabilities in the form of an equation, we could write:

$$P(s'|s,a) = \begin{cases} 0.8 & s' = a, \ \ a \in \text{NEIGHBORS}(s) \\ 0.2 & s' = s \\ 0 & \text{otherwise} \end{cases}$$

After one round of value iteration, $U_1(s) = R(s)$.

(a) After the second round of value iteration, with discount factor $\gamma = 1$, what are the values of all of the states? In other words, what is $U_2(s)$ for each of the six states? List the six values, in left-to-right, top-to-bottom order.

Solution: With states indexed by (row,column), we have:

$$U_2(1,1) = -0.08 = -0.04 + \gamma\max(0.8\cdot(-0.04)+0.2(-0.04),\ 0.8\cdot(-1)+0.2(-0.04))$$
$$U_2(1,2) = 0.752$$
$$U_2(2,1) = -1.0$$
$$U_2(2,2) = 1.0$$
$$U_2(3,1) = 0.0$$
$$U_2(3,2) = 0.752$$

(b) After how many rounds of value iteration (at what value of t) will $U_t(\text{START})$, the value of the starting state, become positive for the first time?

Solution: $U_3(3,1) = 0.6016$, so it first becomes positive at $t = 3$.

Question 19 *(10 points)*

A cat lives in a two-room apartment. It has two possible actions: purr, or walk. It starts in room $s_0 = 1$, where it receives the reward $r_0 = 2$ (petting). It then implements the following sequence of actions: a_0 =walk, a_1 =purr. In response, it observes the following sequence of states and rewards: $s_1 = 2$, $r_1 = 5$ (food), $s_2 = 2$.

(a) (3 points) The cat starts out with a Q-table whose entries are all $Q(s,a) = 0$, then performs one iteration of TD-learning using each of the two SARS sequences described above (one iteration/time step, for two time steps). Because the cat doesn't like to worry about the distant future, it uses a relatively high learning rate ($\alpha = 0.05$) and a relatively low discount factor ($\gamma = \frac{3}{4}$). Which entries in the Q-table have changed, after this learning, and what are their new values?

Solution:

- $t = 0$:

$$Q_{local} = r_0 + \gamma \max_a Q(s_1, a) = 2 + 0 = 2$$
$$\begin{aligned} Q(1,\text{walk}) &\leftarrow Q(1,\text{walk}) + \alpha(Q_{local} - Q(1,\text{walk})) \\ &= 0 + 0.05(2-0) = 0.1 \end{aligned}$$

- $t = 1$:

$$Q_{local} = r_1 + \gamma \max_a Q(s_2, a) = 5 + 0 = 5$$
$$\begin{aligned} Q(2,\text{purr}) &\leftarrow Q(2,\text{purr}) + \alpha(Q_{local} - Q(2,\text{purr})) \\ &= 0 + 0.05(5-0) = 0.25 \end{aligned}$$

So the changed values are $Q(1,\text{walk}) \leftarrow 0.1$ and $Q(2,\text{purr}) \leftarrow 0.25$.

(b) (2 points) Instead of model-free learning, the cat decides to implement model-based learning. It estimates $P(s'|s,a)$ using Laplace smoothing, with a smoothing parameter of $k = 1$, using the two SARS observations listed at the start of this problem. What are the new values of $P(s'|s=2,a=\text{purr})$ for $s' \in \{1,2\}$?

Solution:

$$P(s'=1|s=2,a=\text{purr}) = \frac{1+\text{Count}(s_t=2,a_t=\text{purr},s_{t+1}=1)}{2+\sum_{s'}\text{Count}(s_t=2,a_t=\text{purr},s_{t+1}=s')} = \frac{1}{3}$$
$$P(s'=2|s=2,a=\text{purr}) = \frac{1+\text{Count}(s_t=2,a_t=\text{purr},s_{t+1}=2)}{2+\sum_{s'}\text{Count}(s_t=2,a_t=\text{purr},s_{t+1}=s')} = \frac{2}{3}$$

(c) (3 points) After many rounds of model-based learning, the cat has deduced that $R(1) = 2$, $R(2) = 5$, and $P(s'|s,a)$ has the following table:

a:	purr		walk	
s:	1	2	1	2
$P(s'=1\|s,a)$	2/3	1/3	1/3	2/3
$P(s'=2\|s,a)$	1/3	2/3	2/3	1/3

The cat decides to use policy iteration to find a new optimal policy under this model. It starts with the following policy: $\pi(1) = \text{purr}$, $\pi(2) = \text{walk}$. Now it needs to find the policy-dependent utility, $U^\pi(s)$. Again, because the cat doesn't care about the distant future, it uses a relatively low discount factor ($\gamma = 3/4$). Write two linear equations that can be solved to find the two unknowns $U^\pi(1)$ and $U^\pi(2)$; your equations should have no variables in them other than $U^\pi(1)$ and $U^\pi(2)$.

Solution: The two equations are

$$U^\pi(1) = R(1) + \frac{3}{4}\sum_{s'} P(s'|1,\pi(1))U^\pi(s')$$
$$U^\pi(2) = R(2) + \frac{3}{4}\sum_{s'} P(s'|2,\pi(2))U^\pi(s')$$

Plugging in the given values of all variables, we have

$$U^\pi(1) = 2 + \frac{3}{4}\left(\frac{2}{3}U^\pi(1) + \frac{1}{3}U^\pi(2)\right)$$
$$U^\pi(2) = 5 + \frac{3}{4}\left(\frac{2}{3}U^\pi(1) + \frac{1}{3}U^\pi(2)\right)$$

(d) (2 points) Since it has some extra time, and excellent python programming skills, the cat decides to implement deep reinforcement learning, using an actor-critic algorithm. Inputs are one-hot encodings of state and action. What are the input and output dimensions of the actor network, and of the critic network?

Solution: The actor network takes a state as input, thus its input dimension is 2 (if the input is a one-hot encoding of two states). It computes the probability that any given action is the best action, so its output dimension is 2 (if there are two possible actions). The critic takes, as input, an encoding of the state (two dimensions), and an encoding of the action (two dimensions, if the action is a one-hot encoding of two possible actions), for a total of 4 input dimensions. It computes, as output, a real-valued score $Q(s,a)$, which is a 1-dimensional (scalar) output.

Question 20 ***(0 points)***

What is the optimal policy defined by the Bellman equation?

Solution:

$$\pi^*(s) = \arg\max_a \sum_{s'} P(s'|s,a)U(s')$$

Question 21 ***(0 points)***

When we apply the Q-learning algorithm to learn the state-action value function, one big problem in practice may be that the state space of the problem is continuous and high-dimensional. Discuss at least two possible methods to address this.

Solution:

1. Discretize the state space.
2. Design a lower-dimensional set of discrete features to represent the states.
3. Use a parametric approximator (e.g., a neural network) to estimate the Q function values and learn the parameters instead of directly learning the state-action value functions.

Question 22 ***(0 points)***

In a Markov Decision Process with finite state and action sets, model-based reinforcement learning needs to learn a larger number of trainable parameters than model-free reinforcement learning.

[✓ **True**]

◯ False

Explain:

Solution: Model-based learning needs to learn $P(s'|s,a)$, a set of $N_S^2 N_a$ parameters, where N_s is the number of states, N_a the number of actions. Model-free learning needs to learn $Q(s,a)$, a set of only $N_s N_a$ trainable parameters.

	Simplified GridWorld Column Number	
	1	2
1	-0.04	0.00
2	-0.04	-1.00
3	1.00	-0.04

Question 23 ***(0 points)***

Consider a simplified version of GridWorld, shown above. Assume that the reward for each state, $R(s)$, is known, and is shown in the map above, but that the transition probabilities $P(s'|s,a)$ are not known. The robot starts in the state with $R(s) = 0.00$; if it reaches either the state with $R(s) = 1.00$ or $R(s) = -1.00$, the game ends.

Let the action variable, a, denote the state to which the robot is trying to move. Assume that, from any state s, for any action a, the possible outcomes s' are only the neighboring states or the same state ($s' \in \{s, \text{NEIGHBORS}(s)\}$), but the probabilities of these outcomes are unknown.

The robot performs the following action:

- Starting state s: the state with $R(s) = 0.00$).
- Action a: robot tries to move to the horizontally neighboring state.
- Ending state s': the move is successful.

Given this one training observation, use Laplace smoothing, with a smoothing parameter of $k = 1$, to estimate the value of $P(s'|s,a)$ for this particular combination of (s,a,s').

Solution: The starting state is $s = (1,2)$. According to the problem description, the possible outcomes are known to be $s \in \{(1,1),(1,2),(2,2)\}$, so Laplace smoothing gives:

$$\begin{aligned} P(s' = (1,1)|s = (1,2), a = L) &= \frac{1+k}{1+3k} \\ &= \frac{2}{4} \end{aligned}$$

Question 24 ***(0 points)***

A cat lives in a two-room apartment; its current state is given by the room number it currently occupies ($s \in \{1,2\}$). It has two possible actions: walk, or purr. The cat attempts to determine the optimum policy using Q-learning. It starts out with an empty Q-table ($Q(s,a) = 0$ for all s and a). Starting in state $s_1 = 1$, it receives the following rewards, performs the following actions, and observes the following resulting states:

t	s	R	a	s
1	1	2	purr	1
2	1	2	purr	1

The cat performs one iteration of time-difference Q-learning with each of these two observations, using a learning rate of $\alpha = 0.1$ and a discount factor of $\gamma = 1$.

(a) After these two iterations of Q-learning, what values in the Q-table have changed?

Solution: The cat has only observed the (s,a) combination (1,purr), so the only entry in the Q table that has changed is $Q(1, \text{purr})$.

(b) After these two iterations of Q-learning, what is $Q(1, \text{purr})$?

Solution: Using the formulas

$$\begin{aligned} Q_{\text{local}}(s,a) &= R(s) + \gamma Q_t(s',a) \\ Q_t(s,a) &= Q_{t-1}(s,a) + \alpha \left(Q_{\text{local}}(s,a) - Q_{t-1}(s,a)\right) \end{aligned}$$

We get the following:

$$\begin{aligned} Q_{\text{local}}(1,\text{purr}) &= 2 + 0 \\ Q_1(1,\text{purr}) &= 0 + 0.1\,(2 - 0) = 0.2 \\ Q_{\text{local}}(1,\text{purr}) &= 2 + 0.2 \\ Q_2(1,\text{purr}) &= 0.2 + 0.1\,(2.2 - 0.2) = 0.4 \end{aligned}$$

So the final value is 0.4.

Question 25 ***(0 points)***

~~A robot~~ fire truck is able to manipulate its own horizontal location (D), the angle of its ladder (θ), and the length of its ladder (L). The ladder has a length of L, and an angle (relative to the x axis) of θ ($0 \le \theta \le \frac{\pi}{2}$ radians), so that the position of the tip of the ladder is

$$(x,z) = (D + L\cos\theta, L\sin\theta)$$

(a) What is the dimension of the configuration space of this robot?

Solution: There are three dimensions: D, L, and θ.

(b) The robot must operate between two buildings, positioned at $x = 0$ and at $x = 10$ meters. No part of the robot (neither its base, nor the tip of the ladder) may ever come closer than 1 meter to either building. What portion of configuration space is permitted? Express your answer as a set of inequalities involving only the variables D, L, and θ; the variables x and z should not appear in your answer.

Solution:

$$1 \le D + L\cos\theta \le 9, \quad 1 \le D \le 9$$

(c) The robot's objective is to save a cat from a tree. The cat is at position $(x,z) = (5,5)$. The robot begins at position $(D = 5, L = 3, \theta = 0)$. The final position of the robot depends on how much it costs to raise the ladder by one radian, as compared to the relative cost of extending the ladder by one meter, and the relative cost of moving the truck by one meter. Why?

Solution: Minimum-cost search for the solution will explore steps, in configuration space, that are of equal cost in each direction. The resulting shortest path will depend on the number of steps required to raise the ladder by $\pi/4$ radians, versus shifting the truck by 4m and extending the ladder. Equivalently: if raising the ladder is more expensive, then the truck will move 4m away, and raise the ladder only a little, but if raising the ladder is cheap, then the truck will raise the ladder up to vertical.

Question 26 *(0 points)*

~~A TBR~~ (two-body robot) is a robot with two bodies. Each of the two bodies can move independently; they're connected by a wi-fi link, but there is no physical link. The position of the first body is (x_1, y_1), the position of the second body is (x_2, y_2).

The robots have been instructed to pick up an iron bar. The bar is 10 meters long. Until the robots pick it up, the iron bar is resting on a pair of tripods, 10 meters apart, at the locations $(1,0)$ and $(11,0)$.

(a) Define a notation for the configuration space of a TBR. What is the dimension of the configuration space?

Solution: The configuration space is a 4-dimensional vector space, (x_1, y_1, x_2, y_2).

(b) In order to lift the iron bar, the robot must reach an OBJECTIVE where one of its bodies is at position $(1,0)$ and the other is at position $(11,0)$. In terms of your notation from part (a), specify the OBJECTIVE as a set of points in configuration space. You may specify the OBJECTIVE as a set of discrete points, or as a set of equalities and inequalities.

Solution:

$$\text{OBJECTIVE} = \{(1,0,11,0),(11,0,1,0)\}$$

(c) If the TBR touches the bar (with either of its bodies) at any location other than the endpoints ($(1,0)$ and $(11,0)$), then the bar falls off its tripods. This constitutes a FAILURE. Characterize FAILURE as a set of points in configuration space. You may specify FAILURE as a set of discrete points, or as a set of equalities and inequalities.

Solution: FAILURE is the set of all vectors (x_1, y_1, x_2, y_2) such that

$$1 < x_1 < 11 \text{ and } y_1 = 0$$

or

$$1 < x_2 < 11 \text{ and } y_2 = 0$$