CS 188 Summer 2022 Introduction to Artificial Intelligence Final

- You have approximately 170 minutes.
- The exam is closed book, no calculator, and closed notes, other than two double-sided "crib sheets" that you may reference.
- For multiple choice questions,
  - □ or **[A]** means mark **all options** that apply
  - ○ or **(A)** means mark a **single choice**

First name	
Last name	
SID	
Exam Room	
Name and SID of person to the right	
Name and SID of person to the left	
Discussion TAs (or None)	

Honor code: "As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others."

By signing below, I affirm that all work on this exam is my own work, and honestly reflects my own understanding of the course material. I have not referenced any outside materials (other than two double-sided crib sheets), nor collaborated with any other human being on this exam. I understand that if the exam proctor catches me cheating on the exam, that I may face the penalty of an automatic "F" grade in this class and a referral to the Center for Student Conduct.

Signature: ______________________________

Point Distribution

Q1.	Potpourri	29
Q2.	Bayes Net and Sampling	10
Q3.	College Indecision	14
Q4.	Dynamic Bayes Net	6
Q5.	Fair or Biased?	6
Q6.	Reinforcement Learning	12
Q7.	So Many Derivatives	8
Q8.	Settlers of Catan	15
	Total	100

THIS PAGE IS INTENTIONALLY LEFT BLANK

Q1. [29 pts] Potpourri

(a) True or False

(i) [1 pt] The Value of Perfect Information (VPI) is always non-negative.

● True ○ False

(ii) [1 pt] When we run Q-learning on a fixed dataset of (state, action, next state, reward) tuples once, it will always converge to the optimal Q-values.

○ True ● False

(iii) [1 pt] Iterative deepening search is optimal when all the edge costs are identical.

● True ○ False

(iv) [1 pt] The space complexity of depth-first search (DFS) is $O(bm)$ where b is the branching factor and m is the maximum depth of the search tree.

● True ○ False

(v) [1 pt] When solving an HMM with states S and evidence E, it is possible for variable elimination and the forward algorithm to reach different solutions for $P(S_t|E_{1:t})$ for some timestep t.

○ True ● False

(vi) [1 pt] In a constraint satisfaction problem, if we wanted to prune the domain as much as possible before selecting values, we would use the LCV heuristic.

○ True ● False

Least constraining values heuristic does not prune the domain

(vii) [1 pt] There exists an MDP such that value iteration does not converge for some states but policy iteration converges for all states.

● True ○ False

(viii) [1 pt] Consider a Markov chain with transition probabilities $P(X_t|X_{t-1})$. For two different initial distributions $P(X_0)$, the stationary distributions (if they both exist) are guaranteed to be different.

○ True ● False

Least constraining values heuristic does not prune the domain

(ix) [1 pt] In Bayes Nets, sampling methods usually have smaller memory requirements than exact inference methods (e.g. variable elimination).

● True ○ False

(x) [1 pt] While using Naive Bayes with Laplace smoothing, we pick the value of smoothing strength k based on accuracy on the training set.

○ True ● False

We tune the value of k on the held-out set.

(b) **(i)** [2 pts] Which of the following statements are correct about particle filtering?

■ Both the forward algorithm and particle filtering can be used to calculate (or estimate) the probability $P(X_t|E_{1:t})$ for states X_t and evidence $E_{1:t}$.

□ With particle filtering, we often need more samples than likelihood weighting to achieve the same level of accuracy in the estimations.

■ Particle filtering is often computationally less expensive than the forward algorithm.

□ In particle filtering, after we re-sample the particles, the weights of the particles remain unchanged.

○ None of the above.

(ii) [2 pts] Which of the following are correct expressions?

■ $MEU(e) = \max_a \sum_s P(s \mid e)U(s,a)$

□ $MEU(e,e') = \max_a \sum_s P(s \mid e)P(s \mid e')U(s,a)$

□ $VPI(E' \mid e) = MEU(E' \mid e) - MEU(e)$

□ $VPI(E' \mid e) = MEU(E') - MEU(e)$

○ None of the above

1 is the correct expression for MEU, other options are all incorrect

(c) **(i)** [1 pt] While using Naive Bayes with Laplace smoothing, if the training error is low but validation error is much higher, which of the following should we do?

● Increase k ○ Decrease k

Increasing k reduces overfitting

(ii) [2 pts] When running the Perceptron algorithm, adding a feature to the input of the model will never negatively affect its performance on which of the following datasets?

■ Training set
□ Validation set
□ Test set
○ None of the above

(iii) [2 pts] When using a neural network, if the training error is high, which of the following could help in decreasing the training error?

■ Increase the network's size
□ Train on more data
■ Increase training time
■ Decrease the learning rate

Increasing the size and training time all help with underfitting. Decreasing the learning rate may help if the optimization is oscillating.

(d) Assume that we are in a standard Pacman setting where Pacman's goal is to eat all the food pellets while avoiding ghosts. Answer the following true/false questions.

(i) [1 pt] ● T ○ F The position of the ghosts is part of the minimal state space for this problem. Pacman is trying to avoid the ghosts so the ghost positions are necessary in the state space.

(ii) [1 pt] ○ T ● F There exists a state space formulation with all positive edge weights $> \epsilon > 0$ for some constant ϵ where no heuristic would make A^* search optimal. Can always set the heuristic to be 0 and UCS is optimal.

(e) Arvind goes to the casino one night and is playing the following game: Initially, there is \$1 in a pot. At every round, Arvind has two actions: (1) spend \$$R$ to draw a card from a deck or (2) leave and take the money in the pot (thereby terminating the game). For the draw action, there is a 1/4 chance that the card drawn is a winning card which multiplies the amount of money in the pot by 10, else the money in the pot is reset to \$1 and the game continues. Once the pot reaches \$100, the game ends and Arvind receives \$100 in reward. In all subparts, use a discount value of $\gamma = 1$.

(i) [3 pts] For this part only, let $R = 1$. What are the values of each state in value iteration for times $t = 1$ and $t = 2$? Note that S_i represents the state where the center pot contains \$$i$.

(1) $V_1(S_1) =$ 1 **(2)** $V_1(S_{10}) =$ 10 **(3)** $V_1(S_{100}) =$ 100

(4) $V_2(S_1) =$ 2.25 **(5)** $V_2(S_{10}) =$ 24.75 **(6)** $V_2(S_{100}) =$ 100

(4) $0.25 \cdot 10 + 0.75 \cdot 1 - 1 = 2.25$
(5) $0.25 \cdot 100 + 0.75 \cdot 1 - 1 = 24.75$

(ii) [2 pts] Again using a discount value of $\gamma = 1$, for what value of R will the agent be indifferent between taking the action draw or leave at state S_{10} at time $t = 2$? In other words, determine the value of R where $Q_2(S_{10}, draw) = Q_2(S_{10}, leave)$.

15.75

$Q(S_{10}, draw) = 0.25 \cdot 100 + 0.75 \cdot 1 - R = 25.75 - R$
$Q(S_{10}, leave) = 10$
$25.75 - R = 10 \rightarrow R = 15.75$

(f) **(i)** [1 pt] Which of the following models use a *factored* state representation?

□ Search ■ CSP ■ Bayes Net □ MDP

(ii) [1 pt] Which of the following models assumes *stochastic* transitions?

□ Search □ CSP ■ Bayes Net ■ MDP

(iii) [1 pt] Which of the following models assumes *known physics*?

■ Search ■ CSP ■ Bayes Net ■ MDP

Q2. [10 pts] Bayes Net and Sampling

(a) [2 pts] A, B, C are discrete random variables. Given $A \perp\!\!\!\perp B | C$, which of the following equations must hold?

- ■ $P(A|B,C)P(B|A,C) = P(A,B|C)$
- □ $P(A,B,C) = P(A)P(B)P(A,B|C)$
- ■ $P(A|C) = \frac{P(A)P(C|A)}{P(C)}$
- □ $P(A,B|C) = P(A,B)$
- ○ None of the above.

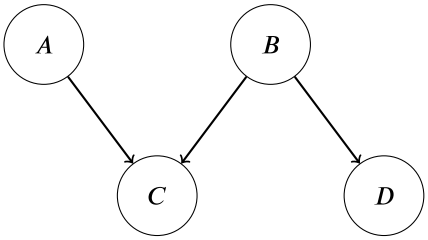
The first option is correct. We can simplify the left side by $P(A|B,C) = P(A|C)$ and $P(B|A,C) = P(B|C)$, then it follows from conditional independence definition.

The second option is incorrect. The left side is equal to $P(A,B|C)P(C)$, so this is true only if $P(A)P(B) = P(C)$ which doesn't make sense.

The third option is correct. It follows from Bayes theorem and doesn't assume any conditions.

The last option is incorrect. The joint probability of A, B can very much depend on C.

Consider the following Bayes Net involving binary random variables A, B, C, D. The relevant probability tables are given.



$P(A)$	
$A = 0$	0.5
$A = 1$	0.5

$P(B)$	
$B = 0$	0.5
$B = 1$	0.5

A	B	C	$P(C\|A,B)$
0	0	0	0.6
0	0	1	0.4
0	1	0	0.4
0	1	1	0.6
1	0	0	0.8
1	0	1	?
1	1	1	0.6
1	1	0	0.4

B	D	$P(D\|B)$
0	0	0.6
0	1	0.4
1	0	0.4
1	1	0.6

(b) **(i)** [1 pt] Calculate $P(C = 1|A = 1, B = 0)$ (the ? entry in the table).

1-0.8=0.2

(ii) [1 pt] Calculate $P(A = 1|B = 0, D = 1)$.

$= P(A = 1) = 0.5$

(c) Instead of calculating the exact quantity, suppose we want to estimate $P(C = 1|D = 1)$ using different sampling methods.

(i) [1 pt] In this subpart we use rejection sampling. Which of the following is a valid topological order and is most efficient for rejection sampling to estimate $P(C = 1|D = 1)$?

- ○ A, B, C, D
- ○ B, A, D, C
- ● B, D, A, C
- ○ D, C, B, A

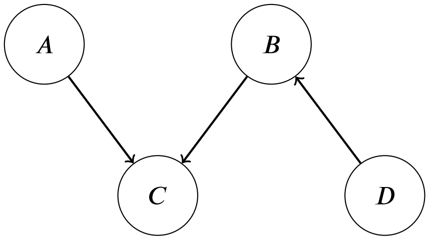
(ii) [1 pt] In this subpart we use likelihood weighting. What is the weight of the sample $(A = 0, B = 0, C = 0, D = 1)$?

0.4

(iii) [2 pts] In this subpart we use Gibbs sampling. We initialize $A = 0$, $B = 0$, $C = 0$, $D = 1$, and choose to re-sample A. What is the probability that we still get $A = 0$ after re-sampling?

$$P(A = 0 | B = 0, C = 0, D = 1) = \frac{P(A=0,B=0,C=0)}{\sum_a P(A=a,B=0,C=0)} = \frac{0.5*0.5*0.6}{0.5*0.5*0.6+0.5*0.5*0.8} = 3/7$$

(d) [2 pts] We reverse the arrow between B and D to create a new Bayes Net (shown below).



Which of the following statements are true?

● The set of joint distributions $P(A, B, C, D)$ that can be modeled by the two Bayes nets are the same.

○ The set of joint distributions that can be modeled by the old Bayes net is a subset of the set of joint distributions that can be modeled by the new Bayes net.

○ The set of joint distributions that can be modeled by the new Bayes net is a subset of the set of joint distributions that can be modeled by the old Bayes net.

○ None of the above.

Q3. [14 pts] College Indecision

(a) Pacman is paying to enter a lottery for summer classes, and his favorite class, CS 188, is among the n possible classes. CS 188 would cost \$10, and all other classes in the lottery cost \$1. Pacman is a rational agent, and his utility function is $U_1(\$x) = x^2$, where x is the cost of the class that Pacman wins in the lottery.

(i) [2 pts] For this subpart only, Pacman would win 1 of n possible classes in the lottery. What is the utility of the lottery? You may leave your answer in terms of n.

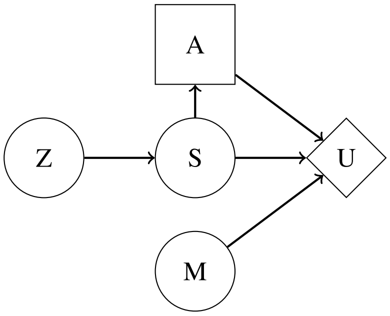
$$\frac{1}{n}\cdot U_1(\$10) + \left(1-\frac{1}{n}\right)\cdot U_1(\$1) = \frac{1}{n}100 + 1 - \frac{1}{n}1 = \frac{1}{n}\cdot 99 + 1$$

(ii) [2 pts] For this subpart only, Pacman would now win 2 of n possible classes in the lottery, and the total utility is the sum of the utility of each class. What is the utility of the lottery? You may leave your answer in terms of n.

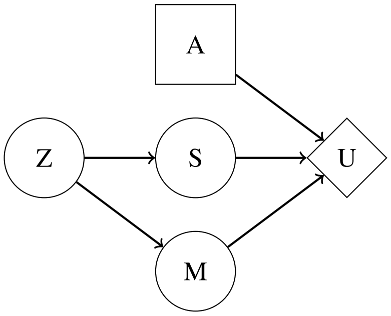
$$\frac{1}{n}\cdot U_1(\$10) + \left(1-\frac{1}{n}\right)\left(\frac{1}{n-1}\right)\cdot U_1(\$1) + \left[1-\frac{1}{n}-\left(1-\frac{1}{n}\right)\left(\frac{1}{n-1}\right)\right]\cdot U_1(\$1)$$
$$= \frac{2}{n}\cdot U_1(\$10) + \left(2-\frac{2}{n}\right)\cdot U_1(\$1)$$
$$= \frac{1}{n}\cdot 200 + 2 - \frac{1}{n}\cdot 2 = \frac{2}{n}\cdot 99 + 2$$

(b) After taking summer classes, Pacman is deciding his major, which depends on what he finds meaningful (M) and how stressed he is (S). The amount he slept the night before (Z) influences how stressed he is.

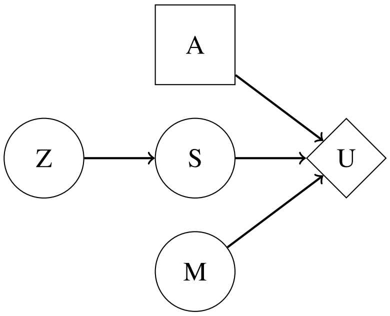
(i) [2 pts] Select all of the decision networks that can represent Pacman's decision.



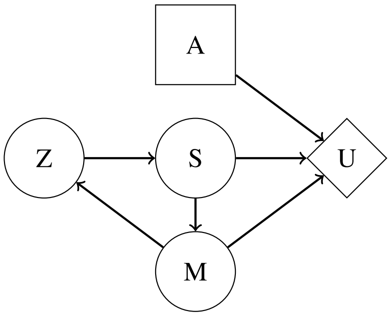
A



B



C



D

☐ A ■ B ■ C ☐ D

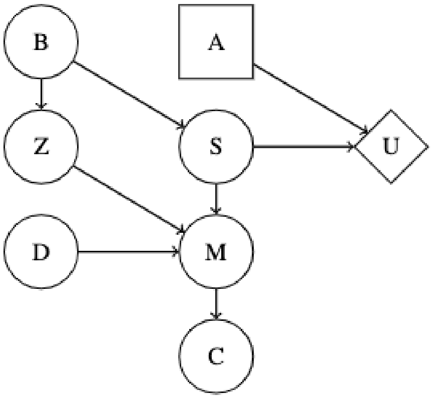
The action should always be completely in our control, so (a) is wrong. (d) has a cycle, so it is not a valid Bayes net. (c) has all the dependencies in the description. (b) has an additional arrow from Z to M, but Z and M could still be independent.

(ii) [3 pts] Pacman keeps track of how long he slept the night before ($Z = z'$). Write out his new MEU as a function of the CPTs corresponding to the decision network.

$EU(a|e) = \sum_s \sum_m P(s, m|z')U(a, s, m) = \sum_s \sum_m P(s|z')P(m)U(a, s, m)$

$\max_a EU(a|e) = \max_a \sum_s \sum_m P(s|z')P(m)U(a, s, m)$

We are now given a completely new decision network as depicted below:



(c) For the following statements, select if they are always, sometimes, or never true.

(i) [1 pt] **VPI(S) ≥ VPI(B)**

● Always true
○ Sometimes true
○ Never true

Since B only affects the utility through S, the VPI of knowing S directly will always be higher than or equal to the VPI of knowing B.

(ii) [1 pt] **VPI(Z|M) + VPI(Z|B) ≤ VPI(Z|B, M)**

○ Always true
● Sometimes true
○ Never true

Given M, there's an active path between Z and the utility node. VPI(Z|B) = 0 because given B but not M, there's no active path between Z and the utility node. Given both, there's the same active path through M.

A case where this is not true: we can construct the CPT for M such that the edge Z-M is disregarded by making $P(M|S, Z, D) = P(M|S, D)$ for all Z. In that case, VPI(Z|B,M) = 0 since the only path Z-B-S is blocked whereas VPI(Z|M) ≥ 0.

(d) [2 pts] Under which of the following conditions (considered independently) can we safely ignore C when solving for the MEU?

■ C is not an evidence variable
□ C is an evidence variable
□ M is not an evidence variable
■ M is an evidence variable
○ None

We can ignore non-evidence leaf nodes, because they do not affect the MEU. Given M, C does not provide us any additional value.

(e) [1 pt] The following question is unrelated to the decision network above. Consider the following lottery $L_1 = [0.5, \$2; 0.5, \$8]$. Indicate whether an agent with utility function $U_2(x)$ is risk-seeking, risk-neutral, or risk-averse on this lottery, where

$$U_2(x) = \begin{cases} 0 & \text{if } x < 0 \\ x^2 & \text{if } 0 \le x \le 6 \\ 4x + 12 & \text{if } x > 6 \end{cases} \tag{1}$$

Recall that a lottery $L = [p_1, P_1; p_2, P_2]$ represents a situation where Pacman receives prize P_1 with probability p_1, and prize P_2 with probability p_2.

○ Risk-seeking

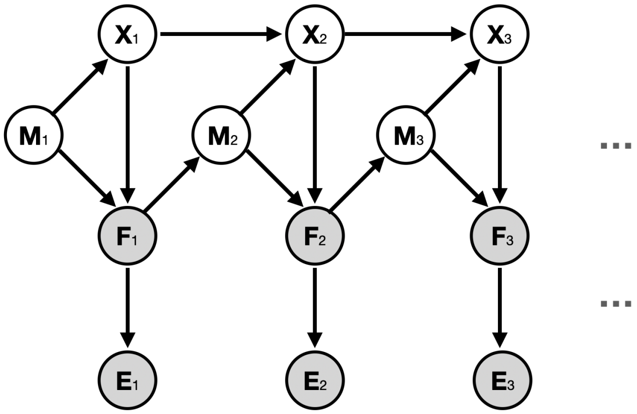
○ Risk-neutral

● Risk-averse

Expected value is $0.5 \cdot \$2 + 0.5 \cdot \$8 = \$5$, which is worth $U_2(\$5) = 25$. $U_2(L_1) = 0.5 \cdot U_2(\$2) + 0.5 \cdot U_2(\$8) = 0.5 \cdot 4 + 0.5 \cdot 44 = 24$, so Pacman does not take the risk here.

Q4. [6 pts] Dynamic Bayes Net

We are given the following dynamic Bayes net:



(a) [2 pts] Which of the following conditional independence relations are correct?

□ $X_{t+1} \perp\!\!\!\perp X_{t-1} | X_t$

□ $F_{t+1} \perp\!\!\!\perp F_{t-1} | F_t$

□ $E_{t+1} \perp\!\!\!\perp E_{t-1} | E_t$

□ $M_{t+1} \perp\!\!\!\perp M_{t-1} | M_t$

● None of the above

Markov property says that future is independent with past given present. From the Bayes net graph we see that given X_i and M_i, which "block" all the path from previous X_{i-1}, M_{i-1} to X_{i+1}, M_{i+1}, X_{i-1}, M_{i-1} will be independent with X_{i+1}, M_{i+1}. However, for F and E we can see that given F_i, there is an active path: F_{i-1}, M_i, X_i, X_{i+1}, F_{i+1}

(b) [2 pts] Which of the following is the correct update rule for the elapse-time (prediction) update?

○ $P(X_t, M_t | f_{1:t-1}, e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}, X_t | f_{1:t-1}, e_{1:t-1}) \sum_{m_{t-1}} P(m_{t-1}, M_t | f_{1:t-1}, e_{1:t-1})$

○ $P(X_t, M_t | f_{1:t-1}, e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}, m_{t-1} | f_{1:t-1}, e_{1:t-1}) P(M_t | f_{t-1}, e_{t-1}) P(X_t | x_{t-1}, Z_{t-1})$

○ $P(X_t, M_t | f_{1:t-1}, e_{1:t-1}) = \sum_{x_{t-1}, m_{t-1}} P(x_{t-1}, m_{t-1} | e_{t-1}, f_{t-1}) P(M_t | f_{t-1}) P(X_t | M_t, x_{t-1})$

● $P(X_t, M_t | f_{1:t-1}, e_{1:t-1}) = \sum_{x_{t-1}, m_{t-1}} P(x_{t-1}, m_{t-1} | e_{1:t-1}, f_{1:t-1}) P(M_t | f_{t-1}) P(X_t | M_t, x_{t-1})$

○ $P(X_t, M_t | f_{1:t-1}, e_{1:t-1}) = \sum_{x_{t-1}, m_{t-1}} P(x_{t-1}, m_{t-1} | e_{1:t-1}, f_{1:t-1}) P(X_t, M_t | m_{t-1}, x_{t-1}) P(X_t | f_{t-1})$

○ $P(X_t, M_t | f_{1:t-1}, e_{1:t-1}) = \sum_{x_{t-1}, m_{t-1}} P(x_{t-1}, m_{t-1} | e_{1:t-1}, f_{1:t-1}) P(m_{t-1} | f_{t-1}) P(X_t | M_t, x_{t-1})$

○ None of the above.

(c) [2 pts] What is the correct update rule for the observation update? From the six options below, select the **minimum set** of options such that, after multiplying them and normalizing, gives $P(X_t, M_t | f_{1:t}, e_{1:t})$.

□ $P(x_{t-1}, m_{t-1} | e_{1:t-1}, f_{1:t-1})$

☑ $P(f_t | X_t, M_t)$

□ $P(e_t | f_t)$

□ $P(M_t | f_{t-1})$

□ $P(X_t | M_t, x_{t-1})$

☑ $P(X_t, M_t | f_{1:t-1}, e_{1:t-1})$

○ None of the above.

Q5. [6 pts] Fair or Biased?

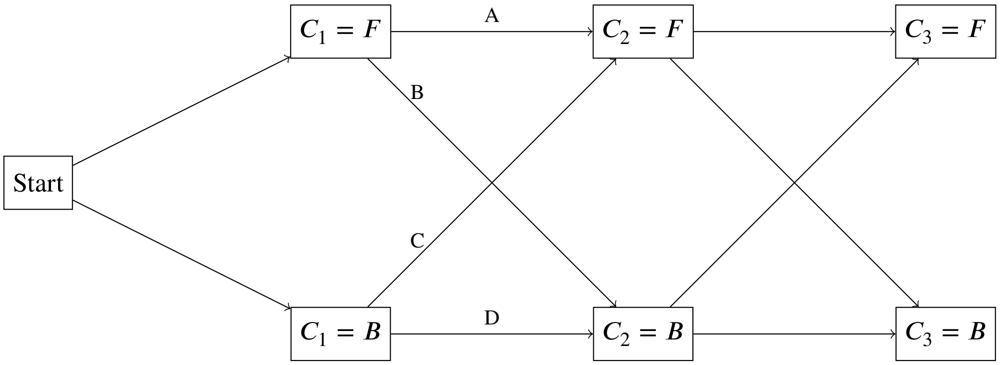
We have two indistinguishable coins. The fair coin (F) has 0.5 probability for getting either heads or tails; the biased coin (B) always gives tails.

Our friend Angela did 3 coin flips, each time with either the fair or the biased coin, but we do not know which. We use C_t ($t = 1, 2, 3$) to represent the coin that Angela chooses for the t^{th} coin flip. We do know, however, that Angela picks the fair or biased coin with equal probability for the first flip (i.e. $P(C_1 = F) = P(C_1 = B) = 0.5$), and chooses the coin at each subsequent timestep according to the following probability table:

C_t	C_{t+1}	$P(C_{t+1}\|C_t)$
F	F	0.75
F	B	0.25
B	F	0.5
B	B	0.5

We also observe the outcomes of the flips to be **(Head, Tail, Tail)**. We want to use the Viterbi algorithm to infer the most likely sequence of coins that Angela picked to flip.

(a) [2 pts] Shown below is the full Trellis Diagram. Fill in the values for each labeled arc connecting C_1 to C_2 with the product of the transition probability and the observation likelihood at the second coin flip (as seen in lecture). Recall that the observation at timestep 1 is **Head** and observation at timestep 2 is **Tail**.



(i) A: 0.375

(ii) B: 0.25

(iii) C: 0.25

(iv) D: 0.5

(b) [2 pts] What is the most likely sequence of coins? Your answer should be a 3-character string, e.g. "BBF" means the first two coin flips are biased while the third is fair. FFF

(c) [2 pts] Which of the following is true about the Viterbi algorithm in general?

☒ The time complexity of Viterbi is linear with regard to the number of time steps.

□ The time complexity of Viterbi is linear with regard to the size of the state space.

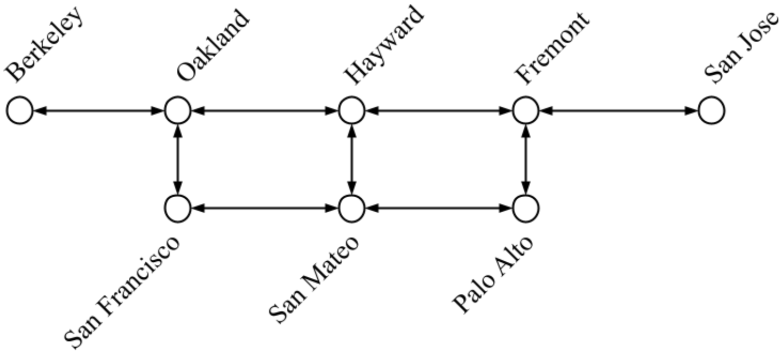
☒ The space complexity of Viterbi is linear with regard to the size of the state space.

□ The Viterbi algorithm computes $\arg\max_{x_{1:N}} P(e_{1:N}|x_{1:N})$, where x are the states and e are the observations.

○ None of the above.

Q6. [12 pts] Reinforcement Learning

(a) In the directed graph below, we formulate the problem of commuting in the Bay Area as a simple MDP, where the cities (nodes) represent the states, and the arrows represent possible actions. We will use the direction of the arrows in the graph, i.e., "up", "down", "left", and "right" to refer to the actions.



(i) [2 pts] Let's assume that the agent does not always succeed in every action, and we want to build an estimate of the transition function $\hat{T}$ and reward function $\hat{R}$ from data (for model-based reinforcement learning). The agent follows some policy π to collect a dataset of (current state, action, next state, reward) tuples, as listed below.

s	**a**	**s'**	**Reward**
San Francisco	up	Oakland	-4
San Francisco	up	San Mateo	-3
San Mateo	up	San Francisco	-5
Oakland	down	San Francisco	-2
San Francisco	up	San Mateo	-3
San Francisco	right	San Mateo	-6

(1) What is $\hat{T}$(San Francisco, up, Oakland)?

$\frac{1}{3}$, since out of the 3 times we take the up action from SF, only once do we actually end up in Oakland.

(2) What is $\hat{R}$(San Francisco, up, Oakland)? -4

(ii) [1 pt] We decide to use temporal difference learning instead to learn the values of π. Assume we start with the following values for each state:

s	Berkeley	Oakland	Hayward	Fremont	San Jose	San Francisco	San Mateo	Palo Alto
$V^{\pi}(s)$	-2	-8	-7	-3	-1	-2	-6	-1

We do one update step with the sample **(San Francisco, up, Oakland, -4)**. Assume discount factor $\gamma = 0.9$ and the learning rate $\alpha = 0.5$. What is the updated value of V^{π}(San Francisco)?

sample $=$ R(San Francisco, up, Oakland) $+ 0.9 \cdot V^{\pi}$(Oakland) $= -4 + 0.9 \cdot -8 = -11.2$. Then, V^{π}(San Francisco) $\leftarrow (1-\alpha)V^{\pi}$(San Francisco) $+ \alpha \cdot$ (sample) $= 0.5 \cdot -2 + 0.5 \cdot -11.2 = -6.6$.

(iii) [2 pts] Which of the following are true about temporal difference learning and Q-learning?

■ Q-learning can learn an optimal policy even if the dataset contains some sub-optimal actions.

□ Q-learning is an on-policy algorithm.

□ Temporal difference learning returns an optimal policy.

■ Temporal difference learning often leads to faster convergence than direct evaluation.

○ None of the above

(1): We still take the max across all valid actions, so taking suboptimal actions doesn't hinder our ability to learn an optimal policy. (2): Q-learning is an off-policy algorithm. (3): We need Q-values to learn an optimal policy, and TD-learning returns values, which aren't sufficient. (4): True.

(iv) [2 pts] We want to use an exploration function to give some preference to visiting less-visited state-action pairs. Which of the following exploration functions f permit this behavior? Assume k is some positive real number, $N(s, a)$ represents the number of times that state-action pair (s, a) has been taken, and ϵ is a very small number (say 0.0001) to avoid division by zero.

- □ $f(s, a) = k \cdot Q(s, a)$
- □ $f(s, a) = k \cdot Q(s, a) \cdot N(s, a)$
- ■ $f(s, a) = Q(s, a) + \frac{k}{N(s,a)+\epsilon}$
- □ $f(s, a) = Q(s, a) + k \cdot N(s, a)$
- ■ $f(s, a) = Q(s, a) + k \cdot e^{-N(s,a)}$
- ○ None of the above

(1): Only Q-states are used (k is a scaling factor applied to all state-action pairs, so it doesn't affect which action is chosen), so exploration isn't encouraged. (2): The value keeps amplifying the more times a route is explored, which will eventually cause only one route to be explored all the time. (3): This creates the desired effect, since as $N(s, a)$ increases, the benefit of exploration (initially k) contracts to 0. (4): Same as (2). (5): The exponential function of a negative exponent creates the same monotonically decreasing effect as (3).

(b) Recall the policy improvement step:

$$\pi_{i+1}(s) = \arg\max_a \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V^{\pi_i}(s')]$$

(i) [2 pts] Which of the following modifications to the reward value $R(s, a, s')$ in the equation above will not affect the policy chosen in policy improvement? Note that we are still using the old unmodified reward $R(s, a, s')$ during policy evaluation. Assume that k is a real number and $k > 0$.

- ■ $R(s, a, s') + k$
- ■ $R(s, a, s') - k$
- □ $k \cdot R(s, a, s')$
- □ $R(s, a, s')/k$
- □ $R(s, a, s')^k$
- ○ None of the above

(1) and (2): Adding or subtracting a positive constant will not affect the relative maximum among possible policies since every original $Q(s, a)$ gets changed to $Q(s, a) \pm k$ for all actions a.

(3) and (4): Consider a deterministic setup with action a_1 that has reward $R(s, a_1, s') = 0$ and $V^{\pi}(s') = 10$ vs. an action a_2 that has reward $R(s, a_2, s') = 2$ and $V^{\pi}(s') = 1$ (note that s' is different in each case since for simplicity of this counterexample we are assuming determinisitic and we are comparing two different actions). Making k a large number would cause $R(s, a_2, s')$ to increase without changing the value of a_1 so for $k > 5$, the policy would change. Same idea for dividing by k would apply for a value of $k = 1/15$. Since we do not modify the reward during policy evaluation, $V^{\pi}(s')$ is consistent with what it was before.

(5) Same argument as above.

(ii) [3 pts] Which of the following expressions below is equivalent to $V^{\pi_i}(s')$? Assume that $0 < \alpha < 1$.

- ■ $Q^{\pi_i}(s', \pi_i(s'))$
- □ $\max_{a'} Q^{\pi_i}(s', a')$
- ■ $\sum_{s''} T(s', \pi_i(s'), s'')[R(s', \pi_i(s'), s'') + \gamma V^{\pi_i}(s'')]$
- □ $\max_{a'} \sum_{s''} T(s', a', s'')[R(s', a', s'') + \gamma V^{\pi_i}(s'')]$
- □ $(1 - \alpha) \cdot V^{\pi_i}(s') + \alpha \cdot [R(s', a', s'') + \gamma V^{\pi_i}(s'')]$
- ○ None of the above

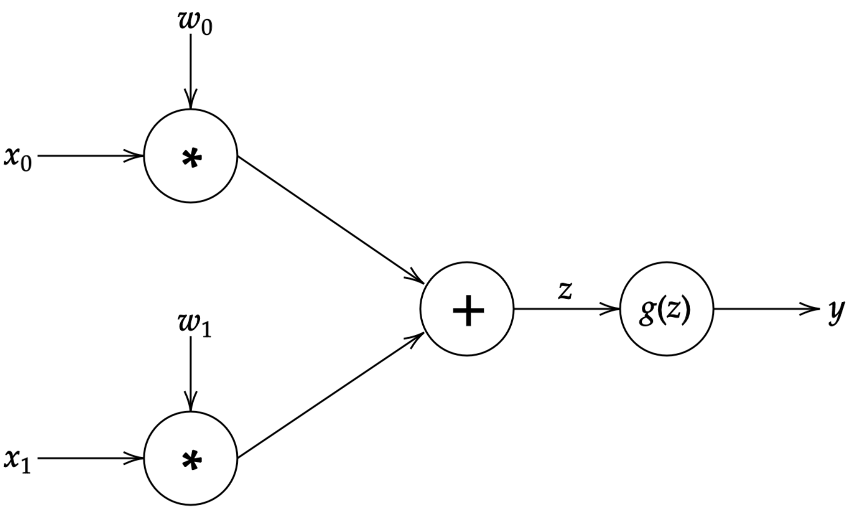
(1) and (3): Correct since these are the equations for policy evaluation.

(2) and (4): Taking the max over actions a' will return the value for the best a', not necessarily the action corresponding to the current policy $\pi_i(s')$, so these are incorrect.

(5): This is the equation for temporal difference learning which is incorrectly used in this context since we do not use a list of samples during policy evaluation for $V^{\pi_i}(s')$. Also, the last term $R(s', a', s'') + \gamma V^{\pi_i}(s'') \neq \sum_{s''} T(s', \pi_i(s'), s'')[R(s', \pi_i(s'), s'') + \gamma V^{\pi_i}(s'')]$

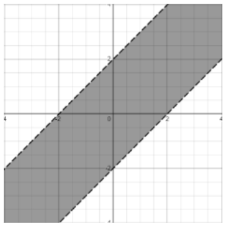
Q7. [8 pts] So Many Derivatives

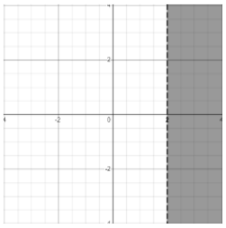
Consider the neural network configuration below.

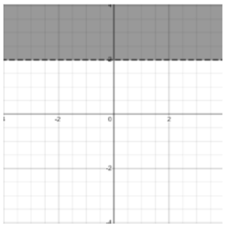


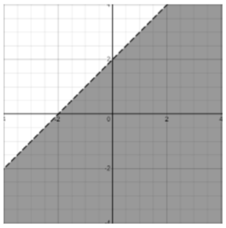
(a) [2 pts] Which of the following decision boundaries can be learned by the neural network? Assume $w_0, w_1, x_0, x_1, T_a \in \mathbb{R}^n$ and $z = w_0 x_0 + w_1 x_1$. Let $g(z)$ be the binary step activation function with T_a as the decision threshold, which is defined as follows:

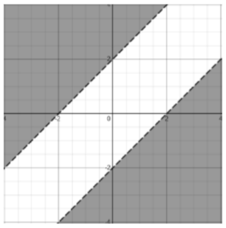
$$g(z) = \begin{cases} 1 \text{ if } z \geq T_a \\ 0 \text{ if } z < T_a \end{cases}$$


A


B


C


D


E

□ Graph A
☒ Graph B
☒ Graph C
☒ Graph D
□ Graph E
○ None of the above

(b) Now let $g(z)$ be the sigmoid activation function and y be a real number value between 0 and 1 (we will ignore the threshold T_a for this part). Recall that the derivative of the sigmoid function is $\frac{\partial}{\partial z} g(z) = g(z) \cdot (1 - g(z))$. You can represent your answers in terms of x_0, x_1, w_0, w_1, z, or y.

(i) [2 pts] Calculate the following partial derivatives for backpropagation.

(1) $\frac{\partial y}{\partial z} = \boxed{y(1-y)}$

(2) $\frac{\partial z}{\partial w_0} = \boxed{x_0}$

(ii) [2 pts] Suppose we are running gradient **descent** on the neural network above. We are trying to minimize the upstream loss L using learning rate α. Given the upstream gradient $\frac{\partial L}{\partial y}$ and the two partial derivatives that you computed in the previous part ($\frac{\partial y}{\partial z}$ and $\frac{\partial z}{\partial w_0}$), determine the gradient descent update rule for w_0.

$$w_0 \leftarrow \boxed{w_0 - \alpha \cdot \frac{\partial L}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w_0}}$$

(c) [2 pts] The Binary Perceptron is defined as the following:

$$y = \text{classify}(x) = \begin{cases} +1 & \text{if } w \cdot f(x) + b \geq 0 \\ -1 & \text{if } w \cdot f(x) + b < 0 \end{cases}$$

where w is a vector of real-valued weights, $w \cdot f(x)$ is the dot product $\sum_{i=1}^{m} w_i f_i(x)$ where m is the number of features, $f_i(x)$ is the ith feature of x, and b is the bias.

Which of the following are true about the binary perceptron as defined above?

□ It is possible that the perceptron learns a decision boundary that is nonlinear in terms of the features $f(x)$.

■ It is possible that the perceptron learns a decision boundary that is nonlinear in terms of the data x.

■ The perceptron algorithm is guaranteed to converge if the data is linearly separable.

□ The perceptron algorithm is trained using gradient descent.

○ None of the above

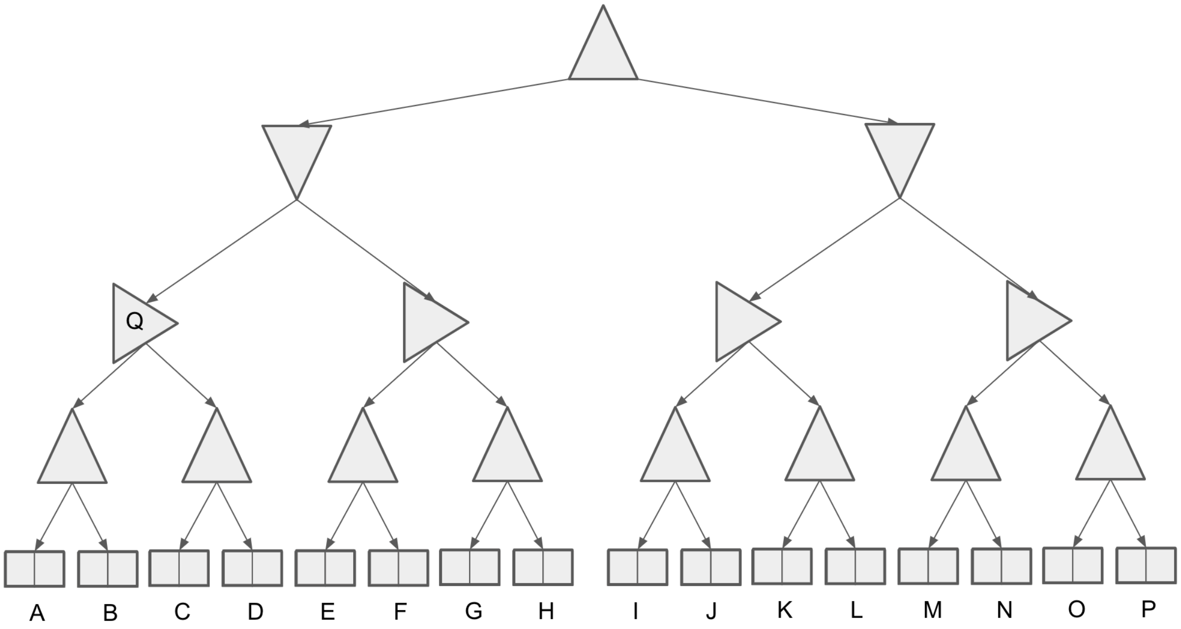
Q8. [15 pts] Settlers of Catan

Sid, Perry, and Andrew are playing a board game. Each terminal state (leaf) has 2 values, x_1 and x_2.

Each player has the following utilities:

- Sid's utility function is $U_s = +x_1$.
- Perry's utility function is $U_p = -x_1$.
- Andrew's utility function is $U_a = x_2 + r \cdot x_1$, where $0 < r < 1$.

Consider the following game tree, with the upward-pointing triangle representing Sid, downwards-pointing triangle representing Perry, and the right-pointing triangle representing Andrew.



(a) [1 pt] This is a zero-sum game. ○ True ● False

(b) Let terminal state values be represented in the form $[x_1, x_2]$. For this part only, assume the leaves $A = [3, 10]$, $B = [4, 5]$, $C = [6, 6]$ and $D = [9, 4]$.

(i) [1 pt] If $r = 0.5$, Which terminal state's values will Q take on? ○ A ○ B ○ C ● D

(ii) [2 pts] What value of r would make Andrew indifferent between choosing left or right at node Q?

$\boxed{5 + 4r = 4 + 9r,\ r = 0.2}$

(c) [4 pts] Assuming the terminal state values are unbounded, mark which leaf nodes can be potentially pruned, under some set of values.

□ A □ B □ C □ D □ E ■ F □ G ■ H □ I □ J □ K □ L
■ M ■ N ■ O ■ P ○ None

A, B, C, and D cannot be pruned because we must establish an option (node Q) for the left minimizer node. F and H can be pruned if E and G, respectively, are worse choices for the minimizer than the value at node Q. I, J, K, and K cannot be pruned by the same reasoning of A, B, C, and D. M, N, O, and P can all be pruned away if the value of the right minimizer node's left child's is worse for the maximizer than the value at the left minimizer node.

(d) [2 pts] Let $[x_{1,i}, x_{2,i}]$ represent the values at leaf node i. When Sid and Perry both act optimally according to their own utilities, for which of the following values of $x_{2,i}$ does Andrew act as a maximizer for Sid? For this part, assume that the first value $x_{1,i}$ for all leaf nodes i is a positive real number.

■ $x_{2,i} = x_{1,i}\ \forall i$

■ $x_{2,i} = x_{1,i}^2\ \forall i$

■ $x_{2,i} = \sqrt{x_{1,i}}\ \forall i$

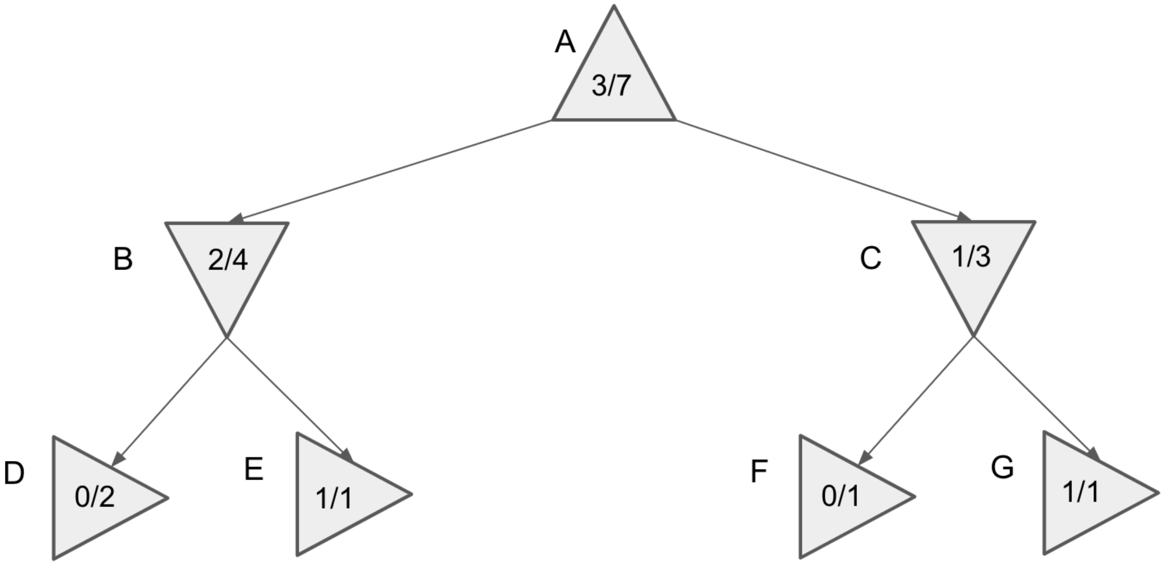
□ $x_{2,i} = \frac{1}{x_{1,i}}\ \forall i$

○ None of the above

Any monotonically increasing function will guarantee that Andrew acts as a maximizer for the higher value of x_1.

(e) Sid decides to run Monte-Carlo Tree Search to train an agent to play for him. He counts the rollouts and the number of favorable outcomes from each node. The current search tree is shown below. Note that the fractions at every node represent the win rates for Sid, the root maximizer. You may assume that Andrew never wins the game, so the complement of Sid's win rate is Perry's win rate.

As a reminder, the UCB heuristic is $UCB(n) = \frac{U(n)}{N(n)} + k\sqrt{log\frac{N(PARENT(n))}{N(n)}}$ where $N(n)$ is the number of rollouts at node n, $PARENT(n)$ is the parent node of n, and $U(n)$ the rollout utility (# wins) for the player at $PARENT(n)$.



(i) [1 pt] If the MCTS algorithm ended now, which action would our agent choose? ● Left ○ Right

(ii) [2 pts] If $k = 188$ in our UCB heuristic, which node will be expanded next?

○ A ○ B ○ C ○ D ○ E ● F ○ G

Since k is so large, UCB will favor the least explored nodes

(iii) [1 pt] In MCTS, a greater k value incentivizes greater exploitation over exploration. ○ True ● False

(iv) [1 pt] When using the UCB heuristic at a node p, if all child nodes of p have the same utility $U(n)$, then the child node with largest $N(n)$ will be expanded in the rollout as long as $k > 0$. ○ True ● False