CS 188 Spring 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,
  - ☐ means mark **all options** that apply
  - ◯ 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 sheet), 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.	Machine Learning	15
Q2.	Satisfiability with a Bayes net	10
Q3.	Ghostbusters 2	13
Q4.	Decisions, Decisions	16
Q5.	She's So Gone	10
Q6.	Search and Games	13
Q7.	Q-Learning on partially observable problems	14
Q8.	Search on MDPs	9
	Total	100

THIS PAGE IS INTENTIONALLY LEFT BLANK

SID: _____

Q1. [15 pts] Machine Learning

(a) First, let's review the basics of perceptrons. The following binary class data has two features, x_1 and x_2. Let $\mathbf{x}$ denote the column vector with these elements. We would like to train a perceptron parameterized by $\mathbf{w} \in \mathbb{R}^2$. The perceptron outputs 1 if $\mathbf{w}^\top \mathbf{x} \geq 0$ and -1 otherwise.

Index	x_1	x_2	Class
1	-1	1	1
2	0	3	1
3	1	-1	-1
4	3	0	-1

(i) [2 pts] If the current weight vector $\mathbf{w} = (-1, -1)$, select the indices of the samples which will be classified correctly.

■ 1 ☐ 2 ☐ 3 ■ 4 ○ None

(ii) [1 pt] If $\mathbf{w}$ is currently all zeros and sample 4 is picked in the first round of training, what will $\mathbf{w}$ be after the update? Assume a learning rate of $\alpha = 1$.

○ (0, 0) ● (-3, 0) ○ (3, 0) ○ (-3, -1) ○ (3, 1) ○ None

(iii) [2 pts] Notice $\mathbf{w}_1 = (-1, 1)$ and $\mathbf{w}_2 = (-4, 1)$ are both perceptrons that can correctly classify all samples in the training data but we still want to figure out how robust they are to unseen data. To do so, we add some noise $\mathbf{n}$ (with magnitude $\|\mathbf{n}\|_2 \leq 1$) to each sample to test the two perceptrons. Which of the two perceptrons is guaranteed to classify the noisy samples correctly?

■ $\mathbf{w}_1$ ☐ $\mathbf{w}_2$ ○ None

(iv) [1 pt] *True/False:* If we add a positive training sample $(2, -1)$, can a linear perceptron correctly classify all the training samples?

○ T ● F

(b) [1 pt] You have trained a perceptron model to perfectly classify your training data but find that it performs poorly on your validation set, which consists of completely new data points. Instead of using the naive perceptron model, which of the following models could improve the validation accuracy?

■ Logistic regression
☐ Linear regression
■ Single-layer neural net with softmax activation
○ None of the above

Logistic regression is an improvement over the perceptron model because it also creates a linear boundary but optimizes the weights w to maximize the likelihood of the training labels given the data points and weights. As a result, it is possible and likely for the classification boundary returned by logistic regression to be more centered and generalizable to new data, allowing it to classify the new validation data better. Since the perceptron model only terminates at the first weight vector that perfectly classifies with no optimization, it is possible that this the perceptron model's classifier is skewed in a particular direction and thus does not perform well on new data.
Linear regression is not used for classification so it cannot be applied to solve this problem.
A single layer neural net with softmax activation is the same as a logistic regression model, so it is true for the same reason that logistic regression is valid.

(c) [2 pts] When using a deep neural network to perform classification, which of the following are learned by the model?

■ Weight matrices of the neural net
■ Bias vectors of the neural net
☐ Hidden layer size of the neural net
☐ Batch size
☐ Learning rate
☐ Number of layers of the neural net
○ None of the above

(d) **(i)** [1 pt] ○ T ● F During training, if a perceptron misclassifies data $x^{(i)}$, it will not misclassify it after the weight update.

(ii) [1 pt] ○ T ● F The number of weights (including bias) of a multiclass logistic regression model with d-dimensional input and 3 classes is $3d + 1$.

(iii) [1 pt] ○ T ● F Applying Laplace smoothing on Naive Bayes could increase validation accuracy because it mitigates underfitting.

(iv) [1 pt] ○ T ● F We cannot use $y = |x|$ as an activation function in neural networks because it's not differentiable at $x = 0$.

(v) [1 pt] ● T ○ F It's possible for a neural network with only 1 hidden layer to represent any continuous function.

(vi) [1 pt] ● T ○ F We should decrease the learning rate if the training loss is going up every epoch.

SID: _____

Q2. [10 pts] Satisfiability with a Bayes net

In lecture, we learned that we can reduce the satisfiability (SAT) problem to inference in a Bayes net, where logical circuits with input variables as parent nodes will output a CNF value as the child node. The Bayes net can have multiple levels to represent nested logical circuits.

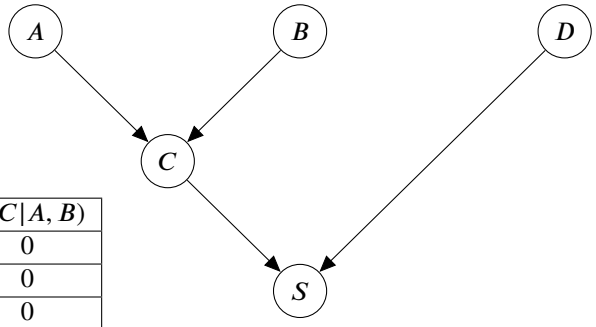
For this problem, we seek to determine the satisfiability of $\neg(A \wedge B) \vee D$ using the following Bayes net.

Here, we construct a Bayes net to represent $C = A \wedge B$ and $S = \neg C \vee D$.

A	$P(A)$
true	0.5
false	0.5

B	$P(B)$
true	0.5
false	0.5

D	$P(D)$
true	0.5
false	0.5



C	A	B	$P(C\|A,B)$
true	*true*	*true*	0
false	*true*	*true*	0
true	*true*	*false*	0
false	*true*	*false*	1
true	*false*	*true*	0
false	*false*	*true*	1
true	*false*	*false*	0
false	*false*	*false*	1

S	C	D	P(S|C,D)
true	*true*	*true*	1
false	*true*	*true*	0
true	*true*	*false*	0
false	*true*	*false*	1
true	*false*	*true*	1
false	*false*	*true*	0
true	*false*	*false*	1
false	*false*	*false*	0

(a) **(i)** [2 pts] Which condition on the probability of S in the Bayes net is true if and only if the logical sentence $\neg(A \wedge B) \vee D$ is satisfiable?

- ○ $P(S = true) = 0$
- ○ $P(S = true) \geq 0$
- ● $P(S = true) > 0$
- ○ $P(S = true) \geq 0.5$
- ○ $P(S = true) > 0.5$
- ○ $P(S = true) = 1$

$P(s) > 0$ iff $\neg(A \wedge B) \vee D$ is satisfiable. Intuitively, you can think of each node in the Bayes net as a bit representing the truth value of its logical statement. You can think of each edge as having positive flow of probability iff there is some variable assignment that makes its source node (not) true. The CPTs act as logical "and" and "or" gates, as well as implementing the "not" operations along the edges. With this construction, each node in the Bayes net has positive probability iff its logical statement is satisfiable.

(ii) [2 pts] Which condition on the probability of S in the Bayes net is true if and only if the logical sentence $\neg(A \wedge B) \vee D$ is valid?

- ○ $P(S = true) = 0$
- ○ $P(S = true) \geq 0$
- ○ $P(S = true) > 0$
- ○ $P(S = true) \geq 0.5$
- ○ $P(S = true) > 0.5$
- ● $P(S = true) = 1$

(b) Now, we will do inference via prior sampling in this Bayes net. Assume that we have access to code that will run prior sampling and return the samples to us. However, *the code will only show us the value of the variable S in each sample*. We can see nothing else.

(i) [1 pt] Suppose we draw 100 samples and find that in exactly 43 of them, S has the value *true*. What can we conclude about the satisfiability of $\neg(A \wedge B) \vee D$?

- ○ $\neg(A \wedge B) \vee D$ is valid.
- ● $\neg(A \wedge B) \vee D$ is satisfiable.
- ○ $\neg(A \wedge B) \vee D$ is **not** satisfiable.
- ○ None of the above.

As long as at least 1 sample has s, we can conclude $P(s) > 0$.

(ii) [1 pt] Suppose we instead draw 100 total samples and find that S has the value *false* in all 100 of them. What can we conclude about the satisfiability of $\neg(A \wedge B) \vee D$?

- ○ $\neg(A \wedge B) \vee D$ is valid.
- ○ $\neg(A \wedge B) \vee D$ is satisfiable.
- ○ $\neg(A \wedge B) \vee D$ is **not** satisfiable.
- ● None of the above.

While it is likely that $P(s) = 0$, we can't be sure. It could be that $P(s) > 0$, but we just got unlucky in all of our samples.

(c) **(i)** [2 pts] Consider the same example of determining satisfiability of $\neg(A \wedge B) \vee D$. If we are still doing prior sampling, what is the probability that our first sample has $S = true$?

$\frac{7}{8}$

There are 8 assignments total (3 variables with 2 possible values each so $2^3 = 8$). Of these assignments, only one of them evaluates to S being false when $A = true$, $B = true$, and $D = false$, which means there are 7 models in total. So the probability of S being true is $\frac{7}{8}$ because the prior is uniform.

(ii) [2 pts] Now consider a sentence with n uniformly distributed binary variables and m models that satisfy the sentence, what is the expected number of samples we draw in order to conclude that the sentence is satisfiable?

Hint: If $X \sim Geometric(p)$, then $\mathbb{E}[X] = \frac{1}{p}$

$\frac{2^n}{m}$

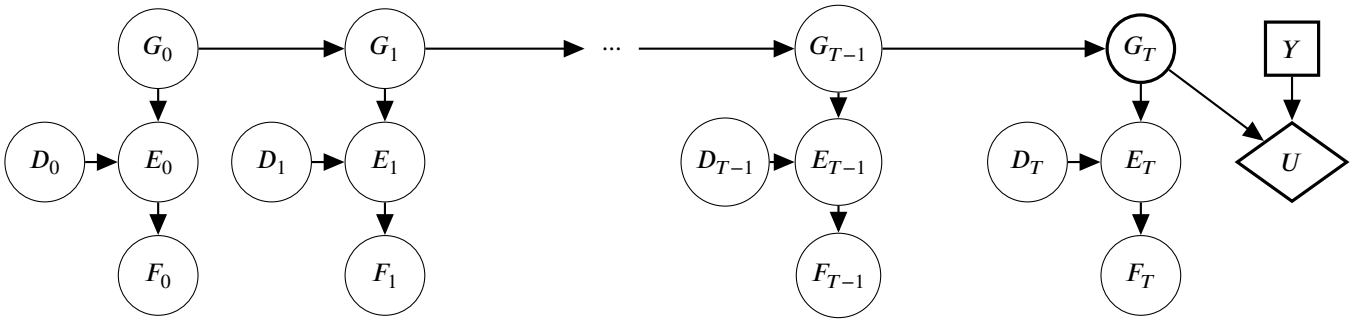
Following the idea from previous part, the probability of drawing a $S = true$ sample will be $\frac{m}{2^n}$. We can conclude as soon as we get a sample that shows $S = true$. Therefore, the number of samples n is a geometric distributed random variable, so $\mathbb{E}[n] = \frac{1}{p} = \frac{2^n}{m}$.

SID: _____

Q3. [13 pts] Ghostbusters 2

A ghost is traveling in an $M \times N$ grid with no walls or obstacles. Pacman is trying to determine the location of the ghost at some finite timestep T, where G_t represents ghost's location at timestep t. At each timestep t, there are up to 3 sensor variables—D_t, E_t, and F_t—connected as shown in the dynamic Bayes net (DBN) below.

Pacman has a guess action Y which represents Pacman's guess of the value G_T. The utility Pacman gains for its guess is determined by $U(Y, G_T)$ which is a predetermined function that is larger when Y is closer to G_T.



(a) [2 pts] For this part only: assume that Pacman does not observe the sensor values and does not know any of the CPTs in the DBN; further, let $U(y, g_T)$ be 100 if $y = g_T$ and -1 otherwise.
Sid from Stanford says the distribution is uniform, assuming that the ghost moves randomly for a long time. Using this information, what is Pacman's maximum expected utility? You may use the values M, N, or T in your expression as necessary.

$MEU(\emptyset) =$ $\frac{100}{MN} - \frac{MN-1}{MN}$

$MEU(\emptyset) = P(Y = G_t)U(Y = G_t) + P(Y \neq G_t)U(Y \neq G_t) = \frac{1}{MN} \cdot 100 + \frac{MN-1}{MN} \cdot (-1) = \frac{100}{MN} - \frac{MN-1}{MN}$.

Since Pacman's guess is random, the chance that Pacman guesses the same square as the ghost's position is $\frac{1}{MN}$.

For the remainder of the question, assume that Pacman knows all the conditional distributions in the DBN.

(b) [1 pt] Which of the following expressions maximizes Pacman's expected utility when no evidence is available?

- ○ $Y = \arg\max_{g_T} P(g_T | G_{T-1})$
- ○ $Y = \arg\max_{g_T} P(g_T | G_{0:T-1})$
- ○ $Y = \arg\max_{g_T} P(G_{0:T-1}, g_T)$
- ● $Y = \arg\max_{g_T} \sum_{g_{T-1}} P(g_T | g_{T-1}) P(g_{T-1})$
- ○ None of the above

We want Y to be the most likely value of G_T, i.e., $\arg\max_{g_T} P(g_T)$ as there is no evidence. By conditioning on g_{T-1}, this is the same as $\sum_{g_{T-1}} P(g_T | g_{T-1}) P(g_{T-1})$. The other expressions actually make no sense, since they involve unobserved random variables.

(c) Pacman can choose to uncover the value of one sensor (seeing its value for all t) in order to help maximize its utility.

(i) [2 pts] Without any assumptions about the sensor CPTs, which of the following inequalities provides the strongest valid statement about the VPIs for the sensor variables?

- ○ $VPI(D) \geq VPI(E) > VPI(F)$
- ○ $VPI(E) \geq VPI(D) > VPI(F)$
- ○ $VPI(E) \geq VPI(F) > VPI(D)$
- ● $VPI(E) \geq VPI(F) \geq VPI(D)$
- ○ Any VPI relationship is possible

$VPI(D)$ is the lowest because D_t and G_t are independent at every timestep t, so $VPI(D) = 0$. $VPI(E) \geq VPI(F)$ since $VPI(E, F) = VPI(E) + VPI(F|E) = VPI(F) + VPI(E|F)$. $VPI(F|E) = 0$ since $F \perp\!\!\!\perp G|E$ and $VPI(E|F) \geq 0$, so we can conclude that $VPI(E) = VPI(F) + VPI(E|F) \geq VPI(F)$.

(ii) [2 pts] Let X and Z be the sensor variables with the highest and second-highest VPI, respectively. Which of the following statements are guaranteed to be true for any timestep $t \in [0, T-1]$ and any values x_t and z_t?

- ☐ $MEU(X_{t+1} = x_{t+1}) \geq MEU(X_t = x_t) \ \forall x_t, x_{t+1}$
- ☐ $MEU(X_{t+1} = x_{t+1}) \geq MEU(Z_t = z_t) \ \forall x_t, z_t$
- ■ $VPI(X_{t+1}) \geq VPI(X_t)$
- ☐ $VPI(X_t) \geq VPI(Z_{t+1})$
- ○ None of the above

It is always possible for the MEU of specific values of a variable X to be less than the MEU of a specific value of Z. $VPI(X) \geq VPI(Z)$ only implies that $MEU(X) \geq MEU(Z)$ but it is not true that for any particular value of X and Z, that $MEU(X = x) \geq MEU(Z = z)$. Therefore, the first two options are not guaranteed to be true.

(iii) [1 pt] Pacman chooses the sensor variable F as evidence, so we are given $F_t = f_t \, \forall t$. Which of the following expressions for Y would maximize Pacman's expected utility?

- ○ $Y = \arg\max_{g_T} \sum_{g_{T-1}} P(g_T|g_{T-1})P(g_{T-1})$
- ○ $Y = \arg\max_{g_T} \sum_{e_T} P(f_T|e_T) \cdot P(g_T|e_{0:T-1})$
- ○ $Y = \arg\max_{g_T} \sum_{e_T} P(f_T|e_T)P(e_T|d_T, g_T) \cdot P(g_T|e_{0:T-1})$
- ● $Y = \arg\max_{g_T} \sum_{e_T, d_T} P(f_T|e_T)P(d_T)P(e_T|d_T, g_T) \cdot P(g_T|f_{0:T-1})$
- ○ None of the above

Choice 4 is the correct expression written using the HMM update equations with the appropriate CPTs.

(iv) [2 pts] Now consider that Pacman instead chooses only the variable E as evidence. Using the functions for the belief distribution (B or B') and the general utility function U, what is the expression for $EU(Y|e_{0:T})$?
Recall that $B(G_T) = P(G_T|e_{0:T})$ and $B'(G_T) = P(G_T|e_{0:T-1})$.

$EU(Y|e_{0:T}) =$ $\sum_{g_T} B(G_T = g_T)U(Y, g_T)$

(d) [3 pts] Pacman is now free to choose as many of the sensor evidence variables as it requires to maximize its expected utility. In addition, Pacman can choose to know the starting position of the ghost. Among all the available evidence listed below, select the minimum set that would allow Pacman to achieve the maximum expected utility possible.

- ■ G_0
- ☐ E_0
- ☐ D_0
- ☐ F_0
- ■ $E_{1:T}$
- ■ $D_{1:T}$

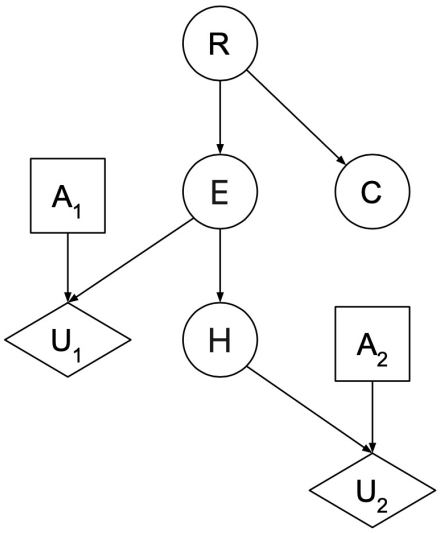
☐ $F_{1:T}$

○ None of the above

The minimum set of variables that would allow Pacman to achieve the maximum expected utility would be the Markov blanket of all the ghost state variables. All variables in the Markov blanket could contribute a positive VPI due to the dependence relationship with G_t and all other variables are independent given the Markov blanket so they would have VPI 0. In this case, the Markov blanket consists of the parent (G_0), children ($E_{1:T}$), and children's parents ($D_{1:T}$) of all the unknown ghost state variables.

Q4. [16 pts] Decisions, Decisions

Saagar plans to attend an event in AILand. His situation can be represented as a decision network with the following components:



Chance variables (all Boolean):

- R: Whether it Rains or not. According to the forecast, $P(r) = 0.3$.
- C: Whether it is Cloudy or not. This may depend on R.
- E: Whether the Event is held or not. This may depend on R.
- H: Whether there is Heavy traffic or not. This may depend on E.

Actions:

- $A_1 \in \{\text{event, arcade}\}$: Whether Saagar goes to the event or arcade.
- $A_2 \in \{\text{car, subway}\}$: Whether Saagar takes a car or subway.

Utilities:

- $U_1(A_1, E)$: The utility Saagar derives from his choice of arcade or event (if it happens).
- $U_2(A_2, H)$: The heavy-traffic-dependent utility Saagar derives from his choice of car or subway.
- Saagar's total utility is the sum: $U = U_1 + U_2$.

It is known that none of the conditional probabilities in the decision network are 0.

(a) [3 pts] Derive an expression for $MEU(R = true)$ in terms of conditional probability and utility terms from the network (no numerical values).

$$MEU(r) = \max_{a_1,a_2} \sum_e P(e|r)[U_1(e, a_1) + \sum_h P(h|e)U_2(h, a_2)]$$

Rubric:
+0.5 Multiplying $P(h|e) * U_2$
+0.5 Summing out over h
+1 Multiplying $P(e|r) * (U_1 + \mathbb{E}(U_2))$
+0.5 Summing out over E
+0.5 Taking max over BOTH actions

(b) [1 pt] Notice that there are two separate utilities U_1 and U_2 that contribute to Saagar's overall utility U. Select all possible individual variables X such that we can correctly decompose $MEU(X = x) = MEU_1(X = x) + MEU_2(X = x)$; i.e., each action can be optimized independently of the other.

■ R ■ C ■ E ■ H ○ None

By linearity of expectation we can always split up the calculation of MEU to maximizing over the two utilities independently.

(c) [1 pt] ● T ○ F Let X be any Boolean variable in any decision network. The following conditions are jointly sufficient to show that $VPI(X) > 0$:

- $P(x)$ is not 0 or 1.
- The optimal decisions given x and $\neg x$ are different.

Now we are given the following numerical values for some of the factors in our decision network:

SID: _____

$P(H|E)$

E	H	$P(H\|E)$
true	*true*	0.8
true	*false*	0.2
false	*true*	0.4
false	*false*	0.6

$U_1(A_1, E)$

A_1	E	$U_1(A_1, E)$
event	*true*	100
arcade	*true*	60
event	*false*	0
arcade	*false*	90

$U_2(A_2, H)$

A_2	H	$U_2(A_2, H)$
subway	*true*	-10
taxi	*true*	-30
subway	*false*	-10
taxi	*false*	0

(d) [2 pts] Given the information we have, which of the following **could** be true?

- [x] $VPI(E) > 0$
- [ ] $VPI(E) = 0$
- [ ] $VPI(E) < 0$
- [x] $VPI(C) > 0$
- [x] $VPI(C) = 0$
- [ ] $VPI(C) < 0$

VPI can never be negative so the last row of options is always false. The VPI for E and C could be positive since E and C both may be dependent with the utility nodes based on the Bayes net. $VPI(C)$ could be 0 because despite the edges in the Bayes Net, C could still be independent of E. Given the table of utilities and probabilities, and knowing that P(E) cannot be 0 or 1 (since P(R) = 0.3 and none of the CPTs have a 0 entry), it is not possible for $VPI(E) = 0$.

(e) [2 pts] Given the information we have, which of the following **must** be true?

- [ ] $VPI(R, C) > VPI(R)$
- [ ] $VPI(R, H) > VPI(E)$
- [x] $VPI(E, H) > VPI(H)$
- [ ] $VPI(C) > VPI(R)$
- ○ None of the above

We want to calculate $VPI(H)$. **For the remainder of this problem**, say we have eliminated R and C to find $P(E = true) = 0.6$.

Say we also have the following values for expected utility given no information.

$EU(A_1, A_2)$

A_1	A_2	$EU(A_1, A_2)$
event	*subway*	50
event	*taxi*	30
arcade	*subway*	40
arcade	*taxi*	20

(f) [2 pts] Find $P(E = true|H = true)$.

$\frac{3}{4}$

$$P(E = true|H = true) = \frac{P(H = true|E = true)P(E = true)}{P(H = true)} \quad (1)$$

$$= \frac{P(H = true|E = true)P(E = true)}{P(H = true|E = true)P(E = true) + P(H = true|E = false)P(E = false)} \quad (2)$$

$$= \frac{0.8 \cdot 0.6}{0.8 \cdot 0.6 + 0.4 \cdot 0.4} = \frac{0.48}{0.64} = \frac{3}{4} \quad (3)$$

(g) [2 pts] Find the expected utility $EU(event, subway|H = true)$. You may show work in the box below for partial credit.

$$EU(event, subway|H = true) = P(E = true|H = true)U_1(event, E = true) +$$
$$P(E = false|H = true)U_1(event, E = false) + U_2(subway, H = true)$$
$$= 3/4 * 100 + 1/4 * 0 + -10 = 65$$

(h) [3 pts] Let's say the best decision if there is traffic is to take subway and go to the event (i.e. $A_1 = event$, $A_2 = subway$). The best decision if there is no traffic is to take a taxi and go to the arcade (i.e. $A_1 = arcade$, $A_2 = taxi$). Here we provide two values that are not calculated from the earlier parts of the question:

$$P(H = false) = 0.36$$
$$EU(arcade, taxi|H = false) = 80\ .$$

Use your answers from the previous part to calculate $VPI(H)$. You may show work in the box below for partial credit.

$$VPI(H) = P(h)MEU(h) + P(\neg h)MEU(\neg h) - MEU(\emptyset)$$
$$= 0.64 * 65 + 0.36 * 80 - 50 = 20.4$$

SID: _____

Q5. [10 pts] She's So Gone

Mrs. Pacman is moving around in a 4x4 gridworld with deterministic transitions and an action space $A = \{up, down, right, left\} = \{\uparrow, \downarrow, \rightarrow, \leftarrow\}$. Any action that would take her into a wall keeps her position constant. There are two possible exiting (absorbing) states: Pacbaby's location at the top right and a fire pit at the bottom right of the gridworld. Exiting from Pacbaby's location gives a reward of +1, while exiting from the fire pit gives a reward of -2. Mrs. Pacman starts in the bottom left corner. The discount factor $\gamma = 1$. Below is a figure of her world:

			+1
Start			-2

(a) We're watching Mrs. Pacman move around and would like to understand how she feels about existing — in other words, we'd like to reason about her *living reward*. Here, we define living reward as a constant reward value that is the same for all non-exit states. For each of the optimal policies displayed below, select the possible living rewards that could have been used to generate each policy. A dot represents the Exit action which is the only action available at exit states.

(i) [2 pts]

→	→	→	·
↑	↑	↑	↑
↑	↑	↑	↑
↑	↑	↑	·

- ☐ -1
- ■ -0.1
- ☐ 0.1
- ☐ 1
- ○ None of the above

(ii) [2 pts]

→	→	→	·
↑	↑	↑	↑
↑	↑	↑	↑
→	→	→	·

- ■ -1
- ☐ -0.1
- ☐ 0.1
- ☐ 1
- ○ None of the above

(iii) [2 pts]

↑	↑	↑	·
↑	↑	↑	↑
↑	↑	↑	↑
←	↑	↑	·

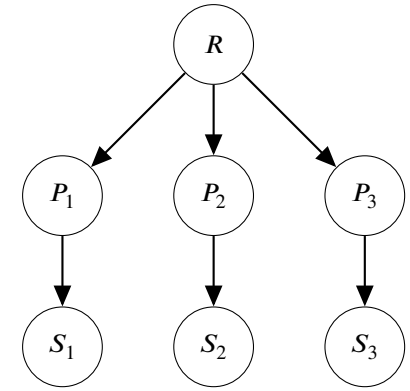
- ☐ -1
- ☐ -0.1
- ☐ 0.1
- ☐ 1
- ● None of the above

(b) [2 pts] Your friend Scott suggests using the Forward Algorithm to estimate Mrs. Pacman's living reward. In the HMM, the hidden variable would be Mrs. Pacman's living reward and the evidence variable would be Mrs. Pacman's observed (true) position. You go to office hours and overhear that this approach wouldn't work. Select all reasons why an HMM model would NOT be a good choice here.

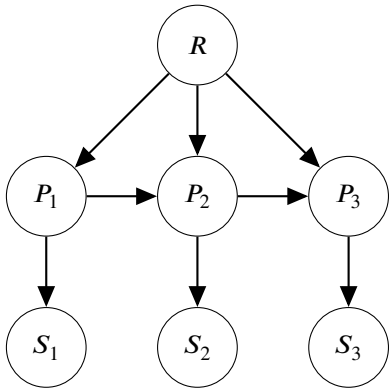
- ■ The evidence variable at a timestep is not independent of everything else given the hidden variable at that timestep
- ■ The hidden variable is not time-varying
- ☐ The discount factor causes the transition model to change between timesteps
- ○ None of the above

(c) [2 pts] Knowing that a vanilla HMM would not be a good model, you decide to consider using a dynamic Bayes net. Select the DBN below that could represent the relationship between Mrs. Pacman's living reward (R), her true position (P_i), and a noisy sensor reading of her location (S_i).

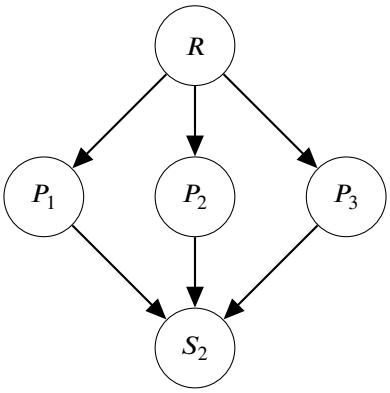
○



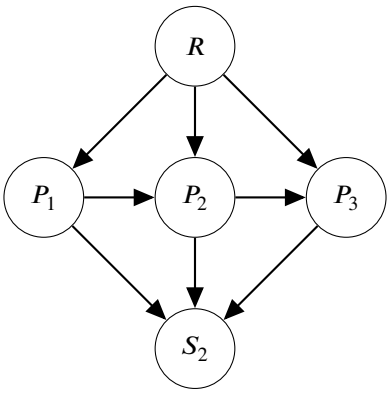
●



○



○



SID: _____

Q6. [13 pts] Search and Games

(a) Please evaluate the truth of each following statement regarding search algorithms and game trees.

(i) [1 pt] ● T ○ F UCS graph search is complete for any graph with edge costs that are strictly greater than some finite positive value ϵ.

(ii) [1 pt] ○ T ● F Greedy graph search is complete if the heuristic is consistent and all edge costs are 1.

(iii) [1 pt] ○ T ● F We can never prune an expectimax tree with bounded leaf nodes.

(iv) [1 pt] ● T ○ F We can never prune a multi-agent non-zero-sum game tree (where each agent has its own independent utility function) with unbounded leaf nodes.

(b) Maria is exploring an M by N grid of the ocean. Her goal is to visit all of the k island squares. There is also one evil pirate ship that will sink Maria's ship if they occupy the same grid square. Maria and the pirate are playing a minimax game where Maria is the maximizer and the pirate is a minimizer. Both agent ships can move up, down, left, right; if they attempt to move off the map, they stay put.

(i) [2 pts] We can view this as a minimax search problem where Maria takes a move and then the pirate ship takes a move. What is the size of the minimal state space in terms of M, N, and k?

$2 \cdot (N \times M)^2 \times 2^k$

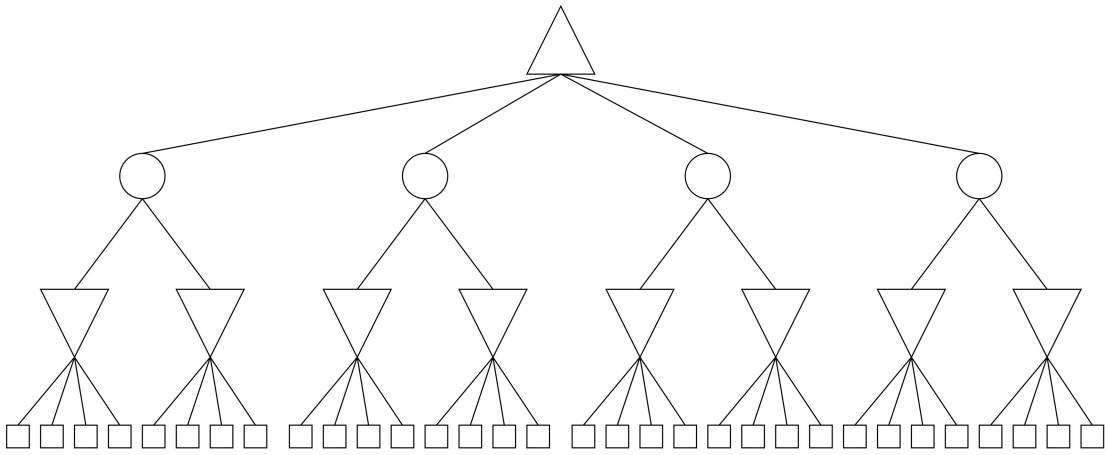
$N \times M$ for each of Maria's grid locations. 2^k to denote the presence of each island times $N \times M$ pirate locations. And 2 to store whether the current move is Maria or the Pirate

(ii) [1 pt] What is the maximum possible branching factor for the minimax tree induced by this search problem?

4

At each timestep, either Maria or the pirate will make a move and they each only have 4 choices.

(c) Suppose now that whenever Maria takes an action, it fails with probability 0.5, resulting in her ship remaining in the same spot. The following game tree depicts the updated scenario. For all pruning subparts below, we do not prune on equality.



(i) [2 pts] What is the maximum number of leaves that can be pruned if there is **no bound** on the leaf values?

9

- Root is a maximizer.
- 1st layer contains 4 expectation nodes (for each of Maria's actions). Each expectation node has two children (for success or not) that points to a minimizer node.
- Second layer contains 8 minimizer nodes. Each has 4 children for 4 possible minimizer actions. Total is 32 leaf nodes.

In the maximal case, the very first expectation node provides the highest value. So first 8 leaves must be seen.
For each of the remaining expectation subtrees, we could have a small value for the left minimizer and after the first

smallest child of the second minimizer ensures the expectation node's value is lower than the first expectation node, we can prune the three remaining leaves of the subtree.

(ii) [2 pts] What is the maximum number of leaves that can be pruned if leaf values are finitely **lower** bounded?

9

Same reason as above since the lower bound does not help with pruning against a root maximizer.

(iii) [2 pts] What is the maximum number of leaves that can be pruned if leaf values are finitely **upper** bounded?

21

In the maximal case, the very first expectation node provides the highest value. So first 8 leaves must be seen.
For the remaining expectation node's subtrees, it is possible for the very first leaf node of the first minimizer to be a very large negative value such that even if all the remaining leaf nodes are the maximal value, the value of the expectation node is lower than the first one. Therefore, we can prune all 7 remaining nodes of the remaining 3 expectimax subtrees.

SID: _____

Q7. [14 pts] Q-Learning on partially observable problems

Pacman is exploring a 3×3 grid world with deterministic transitions and the action space $A = \{up, down, right, left\} = \{\uparrow, \downarrow, \rightarrow, \leftarrow\}$. The states are numbered 1 through 9 as shown below left, and the rewards are as shown below right. The discount factor is $\gamma = 0.5$. **Note that there is no exit action at any state.**

The deterministic transition function $T(s, a)$ returns the result s' of taking action a from state s. (Note: this is different from the $T(s, a, s') = P(s'|s, a)$ notation that we used earlier in the class.) When Pacman hits a wall, it stays in place.

1	2	3
4	5	6
7	8	9

State Numbers

0	0	0
0	0	0
0	0	1

$R(s, a, s')$ for each state s

(a) Let V_2 be the value function after two iterations of value iteration, assuming $V_0(s) = 0$ for all states s.

(i) [2 pts] Write an expression for the policy $\pi_2(s)$ obtained by policy extraction, in terms of T and V_2, for the case where $s \neq 9$.

$\pi_2(s) = \arg\max_a V_2(T(s, a))$

(ii) [2 pts] Select all policies that can be obtained by using policy extraction with V_2 as the value function.

■

→	→	↓
→	↓	↓
→	→	↓

□

→	→	→
↓	↓	↓
→	→	↓

■

↑	→	↓
↓	↓	↓
→	→	↓

○ None

(iii) [2 pts] What is the minimum number of iterations k for value iteration such that the value function V_k is optimal? Write ∞ if the value function is never optimal.

∞

(iv) [2 pts] What is the minimum number of iterations k for value iteration such that the extracted policy π_k is optimal? Write ∞ if the extracted policy is never optimal.

4

For the remainder of the question, we suppose that Pacman only perceives the x-coordinate. In other words, the world from its eyes looks like the 1×3 grid world $\tilde{S} = \{\tilde{1}, \tilde{2}, \tilde{3}\}$ represented below. The action space remains $A = \{up, down, right, left\}$.

$\tilde{1}$	$\tilde{2}$	$\tilde{3}$

(b) [2 pts] What must be accounted for to evaluate the policy learned using Q-learning in this environment? Select all that apply.

- ☐ The problem cannot be represented by a deterministic transition function $T : \tilde{S} \times A \to \tilde{S}$ anymore.
- ■ The problem cannot be represented by a reward function depending only on the state $R : \tilde{S} \to \mathbb{R}$ anymore.
- ■ The problem cannot be represented by a deterministic reward function $R : \tilde{S} \times A \times \tilde{S} \to \mathbb{R}$.
- ☐ The optimal policy cannot be represented by a deterministic function $\pi : \tilde{S} \to A$ anymore.
- ○ None of the above

(c) We modify the reward function of the fully observed grid world as follows.

0	0	0
0	0	1
0	0	0

PacMan still only perceives the x-coordinates.

(i) [2 pts] What must be accounted for to evaluate the policy learned using Q-learning in this environment? Select all that apply.

- ☐ The problem cannot be represented by a deterministic transition function $T : \tilde{S} \times A \to \tilde{S}$ anymore.
- ■ The problem cannot be represented by a reward function depending only on the state $R : \tilde{S} \to \mathbb{R}$ anymore.
- ■ The problem cannot be represented by a deterministic reward function $R : \tilde{S} \times A \times \tilde{S} \to \mathbb{R}$.
- ■ The optimal policy cannot be represented by a deterministic function $\pi : \tilde{S} \to A$ anymore.
- ○ None of the above

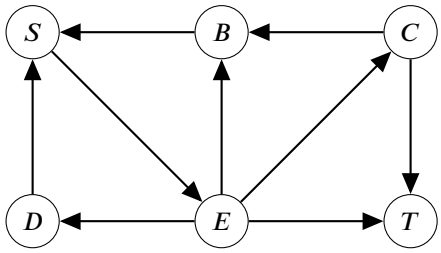
(ii) [2 pts] Assuming an infinite number of samples from each transition and a reasonable decreasing schedule for the learning rate, under which environment will Q-learning always find the optimal policy of the fully observable environment?

- ■ The fully observable environment for part (a)
- ■ The partially observable environment for part (b)
- ☐ The partially observable environment for part (c)
- ○ None of the above

SID: _____

Q8. [9 pts] Search on MDPs

(a) In the following grid, we want to move from start state S to target state T through the shortest possible path. The arrows indicate the available actions. We will formulate the problem as a search problem or an MDP. To formulate it as a search problem, assume an edge cost of 1 for any transition. To formulate it as an MDP, we use a deterministic transition, a reward of 0 when transitioning to the terminal state T, a reward of -1 for any other transition, and a discount factor of 1.



(i) [1 pt] We first try to use uniform-cost tree search. What is total cost of the solution? 2

(ii) [2 pts] We run value iteration to convergence and extract the policy. Which of the following statements are correct?

- [x] The policy at state S will be the same as the action returned by the UCS algorithm at state S.
- [x] Starting at any state, we can go to the terminal state T with the shortest path by following the policy.
- [ ] For the new reward function $R_1(s, a, s') = 1 + R(s, a, s')$, the policy should remain unchanged.
- [x] While standard BFS/UCS tree search no longer works when the transitions are non-deterministic, we can use the same value-iteration/Q-iteration algorithm to generalize to cases where transitions are non-deterministic.
- ○ None of the above

(b) [2 pts] Let's consider the following idea: we first do k iterations of value iteration (with initialization of all zeros), and then use the values $-V_k$ as a heuristic to run A^* search on. Which of the following statements are correct?

- [x] The heuristic is admissible for any k.
- [x] The heuristic is consistent for any k.
- [ ] For $k = 1$, the heuristic will be equivalent to the zero heuristic.
- [x] If $V_k = V^*$, then the number of states expanded by A^* equals the number of states along the optimal path.
- ○ None of the above

(c) In this part, we want to find the shortest path from any state to any state (instead of only to T). To solve this problem in an MDP setting, we define a *goal-conditioned MDP* to be a collection of MDPs with the same state and action space and transition probabilities, but the rewards depend on the goal state. In a goal-conditioned MDP, we define $V^*(s, g)$ as the optimal expected value starting in s when the goal is g, and $Q^*(s, a, g)$ as the optimal value of taking action a in state s when the goal state is g. Also assume that the reward is 0 when we transition to a goal state, and is -1 for all other transitions. The discount factor is 1.

(i) [2 pts] Write down the recursive update equation for $Q_{k+1}(s, a, g)$ assuming that the deterministic transition of taking action a from state s leads to state s' and s' is not a goal state.

$Q_{k+1}(s, a, g) =$ $-1 + \max_{a'} Q_k(s', a', g)$

(ii) [2 pts] Doing Q-learning on a goal-conditioned MDP can require less samples than solving for each MDP with a different goal separately, when the transitional probabilities are unknown (but we know they are the same for all the MDPs with different goals in the same goal-conditioned MDP). Say we have a transition sample (s, a, s'). Write down the Q-learning update rules for a general goal-conditioned MDP with learning rate α and discount factor 1.

$$Q(s, a, g) = \begin{cases} (1-\alpha) \cdot \boxed{Q(s, a, g)} + \alpha \cdot \boxed{\max_{a'} Q(s', a', g) \text{ or } 0} & \text{, if } g = s' \\ (1-\alpha) \cdot \boxed{Q(s, a, g)} + \alpha \cdot \boxed{-1 + \max_{a'} Q(s', a', g)} & \text{, if } g \neq s' \end{cases}$$