CS 188 Spring 2015 — Introduction to Artificial Intelligence — Final V1

- You have approximately 2 hours and 50 minutes.
- The exam is closed book, closed calculator, and closed notes except your three crib sheets.
- Mark your answers ON THE EXAM ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation or show your work.
- For multiple choice questions,
  - ☐ means mark **all options** that apply
  - ◯ means mark a **single choice**
- There are multiple versions of the exam. For fairness, this does not impact the questions asked, only the ordering of options within a given question.

First name	
Last name	
SID	
edX username	

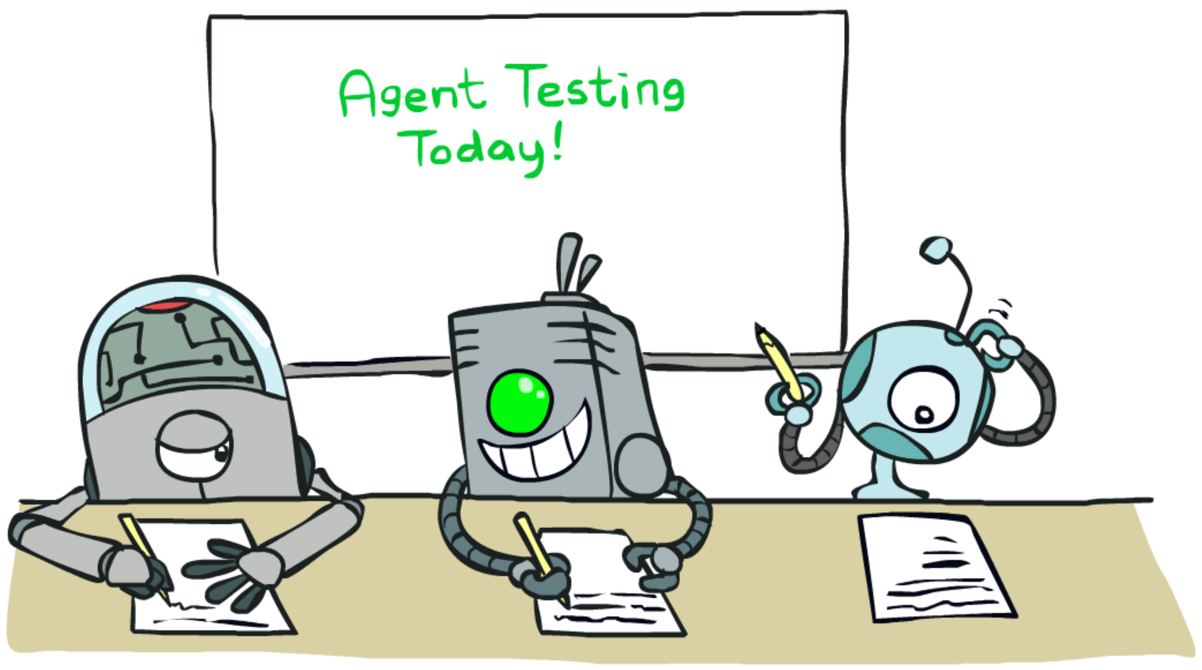
First and last name of student to your left	
First and last name of student to your right	

For staff use only:

Q1.	Agent Testing Today!	/1
Q2.	Power Pellets	/6
Q3.	Satisfying Search	/15
Q4.	Worst-Case Backtracking	/6
Q5.	Best-Case Pruning	/7
Q6.	Ghostbusters	/10
Q7.	MDPs	/6
Q8.	RL	/8
Q9.	Bayes Net and Decision Networks	/8
Q10.	DNA Sequencing	/7
Q11.	Dynamic Bayes' Nets	/11
Q12.	Decision Trees and Other Classifiers	/15
	Total	/100

Q1. [1 pt] Agent Testing Today!

It's testing time! Not only for you, but for our CS188 robots as well! Circle your favorite robot below.



Q2. [6 pts] Power Pellets

Consider a Pacman game where Pacman can eat 3 types of pellets:

- Normal pellets (n-pellets), which are worth one point.
- Decaying pellets (d-pellets), which are worth $max(0, 5 - t)$ points, where t is time.
- Growing pellets (g-pellets), which are worth t points, where t is time.

The pellet's point value stops changing once eaten. For example, if Pacman eats one g-pellet at $t = 1$ and one d-pellet at $t = 2$, Pacman will have won $1 + 3 = 4$ points.

Pacman needs to find a path to win at least 10 points but he wants to minimize distance travelled. The cost between states is equal to distance travelled.

(a) [2 pts] Which of the following must be including for a minimum, sufficient state space?

☐ Location and type of each pellet
☐ Total points Pacman has won
☐ How far Pacman has travelled
☐ Current time
☐ How many pellets Pacman has eaten and the point value of each eaten pellet
☐ Pacman's location
☐ Which pellets Pacman has eaten

(b) [2 pts] Which of the following are admissible heuristics? Let x be the number of points won so far.

☐ Distance to closest pellet, except if in the goal state, in which case the heuristic value is 0.
☐ Distance needed to win $10 - x$ points, determining the value of all pellets as if they were n-pellets.
☐ Distance needed to win $10 - x$ points, determining the value of all pellets as if they were g-pellets (i.e. all pellet values will be t.)
☐ Distance needed to win $10 - x$ points, determining the value of all pellets as if they were d-pellets (i.e. all pellet values will be $max(0, 5 - t)$.
☐ Distance needed to win $10 - x$ points assuming all pellets maintain current point value (g-pellets stop increasing in value and d-pellets stop decreasing in value)
☐ None of the above

(c) [2 pts] Instead of finding a path which minimizes distance, Pacman would like to find a path which minimizes the following:

$$C_{new} = a * t + b * d$$

where t is the amount of time elapsed, d is the distance travelled, and a and b are non-negative constants such that $a + b = 1$. Pacman knows an admissible heuristic when he is trying to minimize time (i.e. when $a = 1, b = 0$), h_t, and when he is trying to minimize distance, h_d (i.e. when $a = 0, b = 1$).
Which of the following heuristics is guaranteed to be admissible when minimizing C_{new}?

☐ $max(h_t, h_d)$ ☐ $min(h_t, h_d)$ ☐ $mean(h_t, h_d)$ ☐ $a * h_t + b * h_d$
☐ None of the above

Q3. [15 pts] Satisfying Search

Consider a search problem $(S, A, Succ, s_0, G)$, where all actions have cost 1. S is the set of states, $A(s)$ is the set of legal actions from a state s, $Succ(s, a)$ is the state reached after taking action a in state s, s_0 is the start state, and $G(s)$ is true if and only if s is a goal state.

Suppose we have a search problem where we know that the solution cost is exactly k, but we do not know the actual solution. The search problems has $|S|$ states and a branching factor of b.

(a) **(i)** [1 pt] Since the costs are all 1, we decide to run breadth-first tree search. Give the tightest bound on the worst-case running time of breadth-first tree search in terms of $|S|$, b, and k.

The running time is $O($ ______________________________ $)$

(ii) [1 pt] Unfortunately, we get an out of memory error when we try to use breadth first search. Which of the following algorithms is the best one to use instead?

○ Depth First Search
○ Depth First Search limited to depth k
○ Iterative Deepening
○ Uniform Cost Search

Instead of running a search algorithm to find the solution, we can phrase this as a CSP:

Variables: $X_0, X_1, X_2, \cdots X_k$

Domain of each variable: S, the set of all possible states

Constraints:

1. X_0 is the start state, that is, $X_0 = s_0$.
2. X_k must be a goal state, that is, $G(X_k)$ has to be true.
3. For every $0 \leq i < k$, (X_i, X_{i+1}) is an edge in the search graph, that is, there exists an action $a \in A(X_i)$ such that $X_{i+1} = Succ(X_i, a)$.

With these constraints, when we get a solution $(X_0 = s_0, X_1 = s_1, \cdots X_k = s_k)$, the solution to our original search problem is the path $s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_k$.

(b) [2 pts] This is a tree-structured CSP. Illustrate this by drawing the constraint graph for $k = 3$ and providing a linearization order. (For $k = 3$, the states should be named X_0, X_1, X_2, and X_3.)

Constraint Graph:

Linearization Order:____

(c) We can solve this CSP using the tree-structured CSP algorithm. You can make the following assumptions:

1. For any state s, computing $G(s)$ takes $O(1)$ time.
2. Checking consistency of *a single arc* $F \to G$ takes $O(fg)$ time, where f is the number of remaining values that F can take on and g is the number of remaining values that G can take on.

Remember that the search problem has a solution cost of exactly k, $|S|$ states, and a branching factor of b.

(i) [1 pt] Give the tightest bound on the time taken to enforce unary constraints, in terms of $|S|$, b, and k.

The running time to enforce unary constraints is $O($ ______________________________ $)$

(ii) [1 pt] Give the tightest bound on the time taken to run the backward pass, in terms of $|S|$, b, and k.

The running time for the backward pass is $O($ ______________________________ $)$

(iii) [1 pt] Give the tightest bound on the time taken to run the forward pass, in terms of $|S|$, b, and k.

The running time for the forward pass is $O($ ______________________________ $)$

(d) [2 pts] Suppose $s_0 \to s_1 \to \cdots \to s_k$ is a solution to the search problem. Mark all of the following options that are *guaranteed* to be true after enforcing unary constraints and running arc consistency.

☐ The remaining values of X_i will be s_i and nothing else.
☐ The remaining values of X_i will be s_i and possibly other values.
☐ A solution can be found by setting each X_i to any of the remaining states in its domain.
☐ A solution can be found by executing the forward pass of the tree-structured CSP algorithm.
☐ None of the above

(e) [4 pts] Suppose you have a heuristic $h(s)$. You decide to add more constraints to your CSP (with the hope that it speeds up the solver by eliminating many states quickly). Mark all of the following options that are valid constraints that can be added to the CSP, under the assumption that $h(s)$ is (a) any function (b) admissible and (c) consistent. *Recall that the cost of every action is* 1.

	Any $h(s)$	$h(s)$ is admissible	$h(s)$ is consistent
For every $0 \le i \le k$, $h(X_i) \le i$	☐	☐	☐
For every $0 \le i \le k$, $h(X_i) \le k - i$	☐	☐	☐
For every $0 \le i < k$, $h(X_{i+1}) \le h(X_i) - 1$	☐	☐	☐
For every $0 \le i < k$, $h(X_{i+1}) \ge h(X_i) - 1$	☐	☐	☐

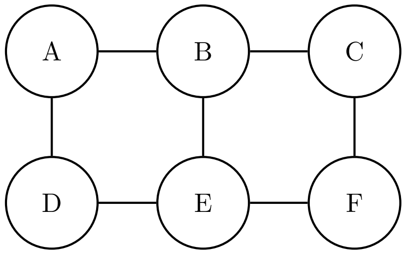
☐ None of the above

(f) [2 pts] Now suppose we only know that the solution will have $\le k$ moves. We do not need to find the optimal solution - we only need to find some solution of cost $\le k$. Mark all of the following options such that if you make single change described in that line it will correctly modify the CSP to find some solution of cost $\le k$. *Remember, the CSP can only have unary and binary constraints.*

☐ Remove the constraints "(X_i, X_{i+1}) is an edge in the search graph". Instead, add the constraints "(X_i, X_{i+1}) is an edge in the search graph, AND $X_i = X_{i+1}$".
☐ Remove the constraints "(X_i, X_{i+1}) is an edge in the search graph". Instead, add the constraints "(X_i, X_{i+1}) is an edge in the search graph, OR $X_i = X_{i+1}$".
☐ Remove the constraint "X_k is a goal state." Instead, add the constraint "There is some i, $0 \le i \le k$, such that X_i is a goal state".
☐ Remove the constraint "X_k is a goal state." Instead, add the constraint "For every $0 \le i \le k$, X_i is a goal state".
☐ None of the above

Q4. [6 pts] Worst-Case Backtracking

Consider solving the following CSP with standard backtracking search where we enforce arc consistency of all arcs before every variable assignment. Assume every variable in the CSP has a domain size $d > 1$.



(a) For each of the variable orderings, mark the variables for which backtracking search (with arc consistency checking) could end up considering more than one different value during the search.

(i) [1 pt] Ordering: A, B, C, D, E, F

☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

(ii) [1 pt] Ordering: B, D, F, E, C, A

☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

(b) Now assume that an adversary gets to observe which variable ordering you are using, and after doing so, gets to choose to add one additional binary constraint between any pair of variables in the CSP in order to maximize the number of variables that backtracking could occur in the worst case. For each of the following variable orderings, select which additional binary constraint should the adversary add. Then, mark the variables for which backtracking search (with arc consistency checking) could end up considering more than one different value during the search when solving the modified CSP.

(i) [2 pts] Ordering: A, B, C, D, E, F

The adversary should add the additional binary constraint:

○ AC ○ AE ○ AF ○ BD
○ BF ○ CD ○ CE ○ DF

When solving the modified CSP with this ordering, backtracking might occur at the following variable(s):

☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

(ii) [2 pts] Ordering: B, D, F, E, C, A

The adversary should add the additional binary constraint:

○ AC ○ AE ○ AF ○ BD
○ BF ○ CD ○ CE ○ DF

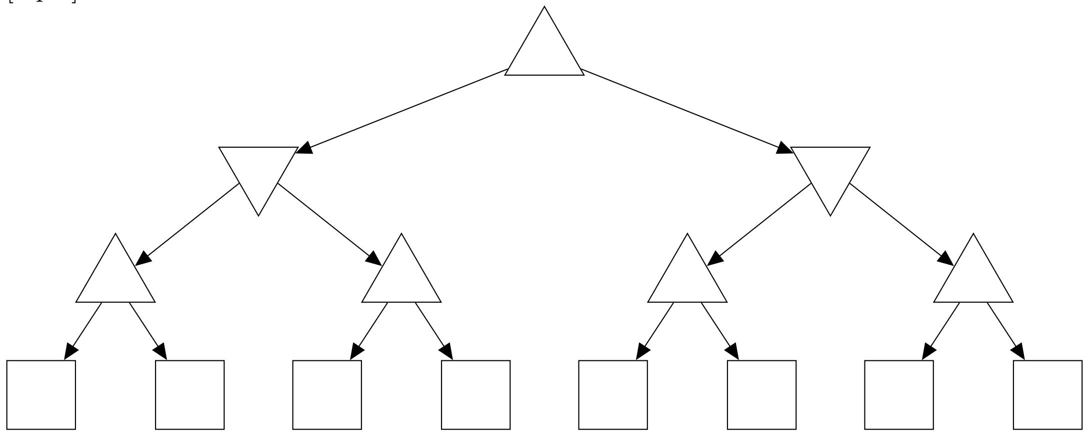
When solving the modified CSP with this ordering, backtracking might occur at the following variable(s):

☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

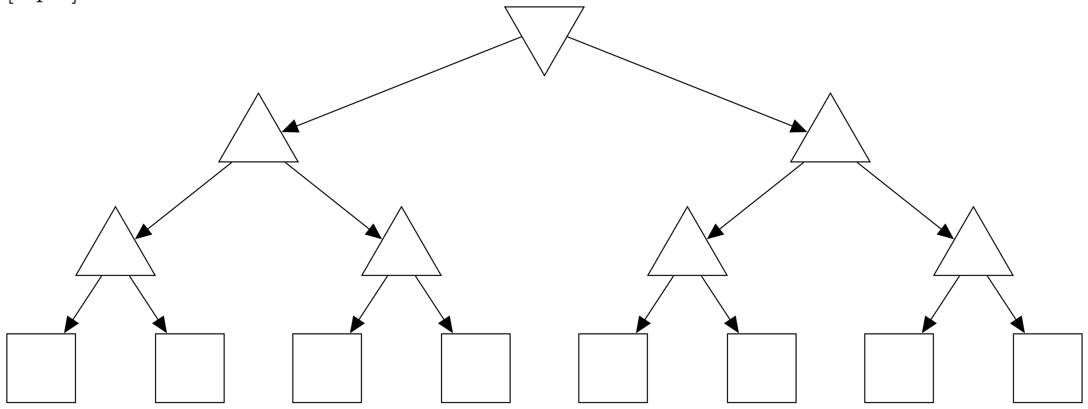
Q5. [7 pts] Best-Case Pruning

For the following zero-sum game trees, the upward pointing triangles represent maximizer nodes, and the downward pointing triangles represent minimizer nodes. Assume that we expand the children of each node in the game tree from left to right. For each tree, cross out the maximal set of **leaf** utility nodes (represented by squares) that can possibly be pruned with a single assignment of the utility nodes, in order to determine the correct minimax value of the root of the game tree. You do *not* need to provide such assignment of the utility nodes.

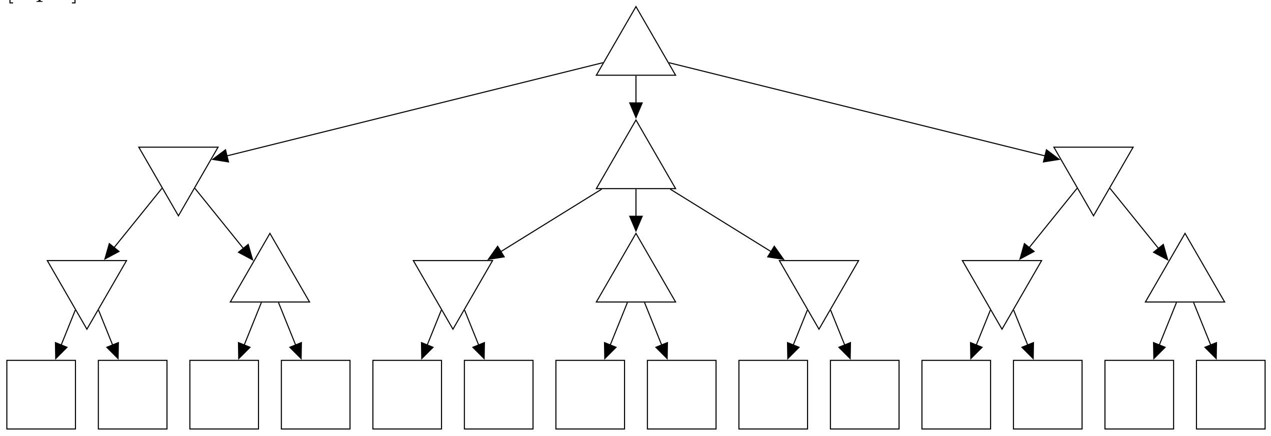
(a) [2 pts]



(b) [2 pts]



(c) [3 pts]



Q6. [10 pts] Ghostbusters

Suppose Pacman gets a noisy observation of a ghost's location for T moves, and then may guess where the ghost is at timestep T to eat it. To model the problem, you use an HMM, where the ith hidden state is the location of the ghost at timestep i and the ith evidence variable is the noisy observation of the ghost's location at time step i. *Assume Pacman always acts rationally.*

(a) [2 pts] If Pacman guesses correctly, he gets to eat the ghost resulting in a utility of 20. Otherwise he gets a utility of 0. If he does not make any guess, he gets a utility of 0.

Which of the following algorithms could Pacman use to determine the ghost's most likely location at time T? (Don't worry about runtime.)

☐ Variable elimination on the Bayes Net representing the HMM
☐ Particle filtering with a lot of particles
☐ Viterbi
☐ Forward algorithm for HMMs
☐ None of the above, Pacman should use ____________________

(b) [2 pts] In the previous part, there was no penalty for guessing. Now, Pacman has to *pay* 10 *utility* in order to try to eat the ghost. Once he pays, he still gets 20 utility for correctly guessing and eating the ghost, and 0 utility for an incorrect guess. Pacman determines that the most likely ghost location at time T is (x, y), and the probability of that location is p.

What is the expected utility of guessing that the ghost is at (x, y), as a function of p? ____________________

When should Pacman guess that the ghost is at (x, y)?

○ Never (he should not guess)
○ If $p <$ ____________________.
○ If $p >$ ____________________.
○ Always

(c) [2 pts] Now, in addition to the -10 utility for trying to eat the ghost, Pacman can also pay 5 utility to learn the exact location of the ghost. (So, if Pacman pays the 5 utility and eats the ghost, he pays 15 utility and gains 20 utility for a total of 5 utility.)

When should Pacman pay the 5 utility to find the exact ghost location?

○ Never
○ If $p <$ ____________________.
○ If $p >$ ____________________.
○ Always

(d) Now, Pacman can try to eat one out of Blinky (B), Inky (I) and Clyde (C) (three of the ghosts). He has some preferences about which one to eat, but he's afraid that his preferences are not rational. Help him out by showing him a utility function that matches his listed preferences, or mark "Not possible" if no rational utility function will work. You may choose any real number for each utility value. **If "Not possible" is marked, we will ignore any written utility function.**

(i) [2 pts] The preferences are $B \prec I$ and $I \prec C$ and $[0.5, B; 0.5, C] \prec I$

$U(B)$	$U(I)$	$U(C)$

○ Not possible

(ii) [2 pts] The preferences are $I \prec B$ and $[0.5, B; 0.5, C] \prec C$ and $[0.5, B; 0.5, C] \prec [0.5, B; 0.5, I]$

$U(B)$	$U(I)$	$U(C)$

○ Not possible

Q7. [6 pts] MDPs

Pacman is in the 3x3 gridworld shown below. In each grid cell, Pacman has 5 actions available: [$\uparrow$, $\downarrow$, $\leftarrow$, $\rightarrow$, $\circ$]. Taking the $\circ$ action moves Pacman to a special **Done** state and ends the game. All actions are deterministic. *Pacman is not allowed to take an action into the wall.* Otherwise, *all actions* (including $\circ$) are available from *all grid cells.*

For each policy, mark the reward function/discount factor pairs for which the policy is optimal.

1. $R_1(s,a,s') = \begin{cases} 1 & s=(0,0), a=\circ, s'=\textbf{Done} \\ 0 & else \end{cases}$
2. $R_2(s,a,s') = \begin{cases} 1 & s=(0,0) \\ 0 & else \end{cases}$
3. $R_3(s,a,s') = \begin{cases} 2 & s'=\textbf{Done} \\ 1 & else \end{cases}$
4. $R_4(s,a,s') = \begin{cases} -3 & a=\circ \\ -1 & else \end{cases}$

Hint: for any $x \in \mathbb{R}$, $|x| < 1$, we have $1 + x + x^2 + x^3 + x^4 + \cdots = 1/(1-x)$.

(a) [2 pts]

	0	1	2
0	$\circ$	$\leftarrow$	$\leftarrow$
1	$\uparrow$	$\uparrow$	$\uparrow$
2	$\uparrow$	$\uparrow$	$\uparrow$

☐ R_1, $\gamma = 0.5$ ☐ R_1, $\gamma = 0.9$
☐ R_2, $\gamma = 0.5$ ☐ R_2, $\gamma = 0.9$
☐ R_3, $\gamma = 0.5$ ☐ R_3, $\gamma = 0.9$
☐ R_4, $\gamma = 0.5$ ☐ R_4, $\gamma = 0.9$
☐ None of the provided options

(b) [2 pts]

	0	1	2
0	$\rightarrow$	$\leftarrow$	$\leftarrow$
1	$\uparrow$	$\uparrow$	$\uparrow$
2	$\uparrow$	$\uparrow$	$\uparrow$

☐ R_1, $\gamma = 0.5$ ☐ R_1, $\gamma = 0.9$
☐ R_2, $\gamma = 0.5$ ☐ R_2, $\gamma = 0.9$
☐ R_3, $\gamma = 0.5$ ☐ R_3, $\gamma = 0.9$
☐ R_4, $\gamma = 0.5$ ☐ R_4, $\gamma = 0.9$
☐ None of the provided options

(c) [2 pts]

	0	1	2
0	$\rightarrow$	$\rightarrow$	$\downarrow$
1	$\uparrow$	$\leftarrow$	$\downarrow$
2	$\uparrow$	$\leftarrow$	$\leftarrow$

☐ R_1, $\gamma = 0.5$ ☐ R_1, $\gamma = 0.9$
☐ R_2, $\gamma = 0.5$ ☐ R_2, $\gamma = 0.9$
☐ R_3, $\gamma = 0.5$ ☐ R_3, $\gamma = 0.9$
☐ R_4, $\gamma = 0.5$ ☐ R_4, $\gamma = 0.9$
☐ None of the provided options

Q8. [8 pts] RL

Pacman is in an unknown MDP where there are three states [A, B, C] and two actions [Stop, Go]. We are given the following samples generated from taking actions in the unknown MDP. For the following problems, assume $\gamma = 1$ and $\alpha = 0.5$.

(a) We run Q-learning on the following samples:

s	**a**	**s'**	**r**
A	Go	B	2
C	Stop	A	0
B	Stop	A	-2
B	Go	C	-6
C	Go	A	2
A	Go	A	-2

What are the estimates for the following Q-values as obtained by Q-learning? All Q-values are initialized to 0.

(i) [2 pts] $Q(C, Stop) =$ ______________________

(ii) [2 pts] $Q(C, Go) =$ ______________________

(b) For this next part, we will switch to a feature based representation. We will use two features:

- $f_1(s, a) = 1$
- $f_2(s, a) = \begin{cases} 1 & a = \text{Go} \\ -1 & a = \text{Stop} \end{cases}$

Starting from initial weights of 0, compute the updated weights after observing the following samples:

s	**a**	**s'**	**r**
A	Go	B	4
B	Stop	A	0

What are the weights after the first update? (using the first sample)

(i) [1 pt] $w_1 =$ ______________________

(ii) [1 pt] $w_2 =$ ______________________

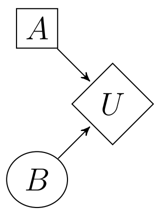
What are the weights after the second update? (using the second sample)

(iii) [1 pt] $w_1 =$ ______________________

(iv) [1 pt] $w_2 =$ ______________________

Q9. [8 pts] Bayes Net and Decision Networks

(a) [2 pts] We have the following decision network with the conditional probability and utility tables:



B	$P(B)$
0	x
1	$1-x$

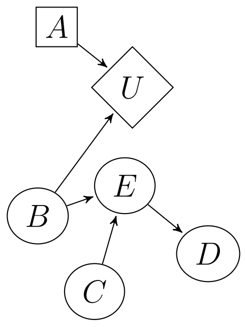
B	A	$U(B,A)$
0	$+a$	10
0	$-a$	4
1	$+a$	0
1	$-a$	8

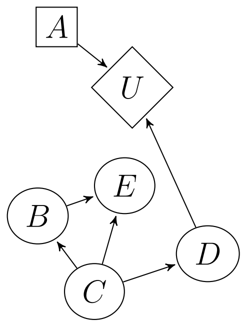
Suppose that we also know that $MEU(B) = 8.5$. What is the value of x?

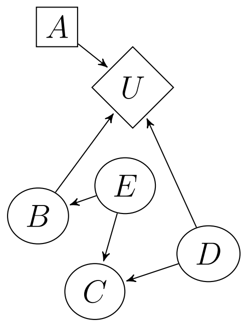
$x =$ ________________________

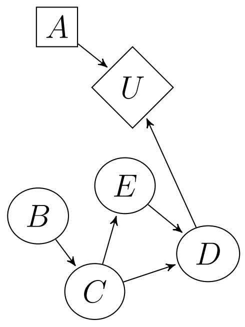
(b) Which of the following decision networks can simultaneously satisfy all of the given VPI and conditional independence constraints for some setting of conditional probability and utility tables? Mark the box below each decision network that can satisfy the constraints, or mark None of the above if none of the decision networks can satisfy the constraints.

(i) [3 pts] $VPI(E) > 0, \quad E \perp\!\!\!\perp C$





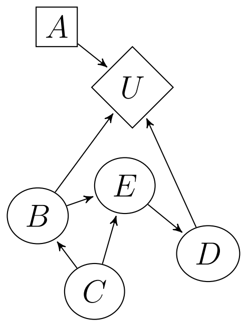


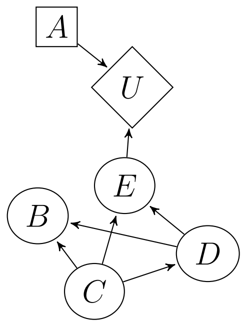


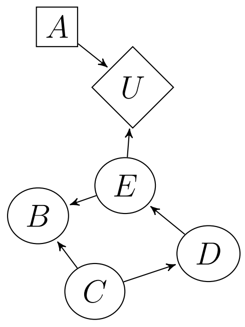
☐ ☐ ☐ ☐

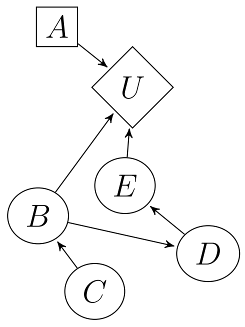
☐ None of the above

(ii) [3 pts] $VPI(C) > 0, \quad VPI(D|E) = 0, \quad C \perp\!\!\!\perp D, \quad C \perp\!\!\!\perp E|B$









☐ ☐ ☐ ☐

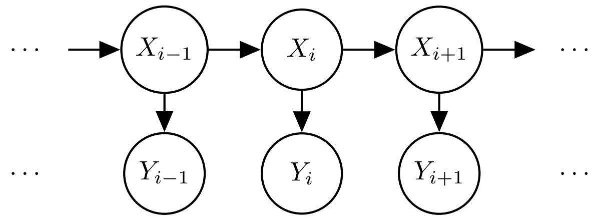
☐ None of the above

Q10. [7 pts] DNA Sequencing

Suppose you want to model the problem of DNA sequencing using the following set-up:

- $X_i, Y_i \in \{A, T, C, G\}$
- X_i : ith base of an individual
- Y_i : ith base output by DNA sequencer

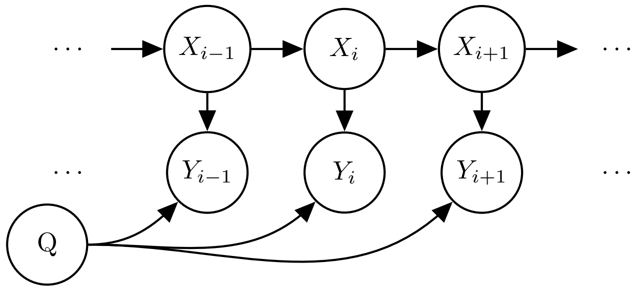
(a) First, you start by using a standard HMM model, shown below.



(i) [1 pt] Which of the following assumptions are made by the above HMM model

☐ $X_i \perp\!\!\!\perp Y_{i+1} \mid X_{i+1} \;\; \forall\, i$
☐ $X_i \perp\!\!\!\perp X_j \;\; \forall\, i \neq j$
☐ $Y_i \perp\!\!\!\perp Y_j \;\; \forall\, i \neq j$
☐ $X_{i-1} \perp\!\!\!\perp X_{i+1} \mid X_i \;\; \forall\, i$
☐ $X_i \perp\!\!\!\perp Y_j \;\; \forall\, i \neq j$
☐ None of the provided options.

(b) Now you want to model the quality of your sequencer with a random variable Q, and decide to use the following modified HMM:



(i) [2 pts] Which of the following assumptions are made by the above modified HMM model?

☐ $X_{i-1} \perp\!\!\!\perp X_{i+1} \mid X_i \;\; \forall\, i$
☐ $X_i \perp\!\!\!\perp X_j \;\; \forall\, i \neq j$
☐ $X_i \perp\!\!\!\perp Y_j \;\; \forall\, i \neq j$
☐ $Y_i \perp\!\!\!\perp Y_j \;\; \forall\, i \neq j$
☐ $X_i \perp\!\!\!\perp Y_{i+1} \mid X_{i+1} \;\; \forall\, i$
☐ $Q \perp\!\!\!\perp X_i \mid Y_i \;\; \forall\, i$
☐ $Q \perp\!\!\!\perp X_i \mid Y_1, ...Y_N \;\; \forall\, i$
☐ $Q \perp\!\!\!\perp X_i \;\; \forall\, i$
☐ None of the provided options.

(ii) [2 pts] You observe the sequencer output $y_1, \ldots, y_N$ and want to estimate probability distribution of the particular sequence of length c starting at base k: $P(X_k \ldots X_{k+c-1} \mid y_1, \ldots y_N)$.
Select all elimination orderings which are maximally efficient with respect to the sum of the generated factors' sizes.

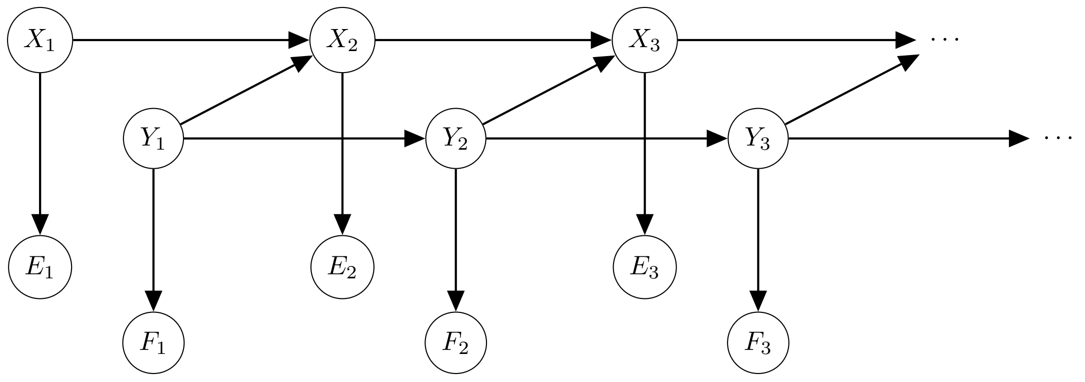
☐ $X_1, \ldots, X_{k-1}, X_{k+c}, \ldots, X_N, Q$
☐ $Q, X_1, \ldots, X_{k-1}, X_N, \ldots, X_{k+c}$
☐ $X_1, \ldots, X_{k-1}, Q, X_{k+c}, \ldots, X_N$
☐ $X_1, \ldots, X_{k-1}, X_N, \ldots, X_{k+c}, Q$
☐ $Q, X_1, \ldots, X_{k-1}, X_{k+c}, \ldots, X_N$
☐ $X_1, \ldots, X_{k-1}, Q, X_N, \ldots, X_{k+c}$
☐ None of the provided options: ____________

(iii) [2 pts] How many entries are in the final conditional probability table $P(X_k, \ldots, X_{k+c-1} \mid y_1, \ldots, y_N)$? The answer takes the form a^b – what are a and b?

$a =$ ________________________________ $b =$ ________________________________

Q11. [11 pts] Dynamic Bayes' Nets

Suppose you have the following Dynamic Bayes Net model, with the associated conditional probability tables (CPTs).



X_1	$\mathbf{P}(X_1)$
$+x_1$	0.5
$-x_1$	0.5

Y_1	$\mathbf{P}(Y_1)$
$+y_1$	0.5
$-y_1$	0.5

X_t	E_t	$\mathbf{P}(E_t \mid X_t)$
$+x_t$	$+e_t$	0.2
$+x_t$	$-e_t$	0.8
$-x_t$	$+e_t$	0.5
$-x_t$	$-e_t$	0.5

Y_t	F_t	$\mathbf{P}(F_t \mid Y_t)$
$+y_t$	$+f_t$	0.4
$+y_t$	$-f_t$	0.6
$-y_t$	$+f_t$	0.8
$-y_t$	$-f_t$	0.2

X_t	Y_t	X_{t+1}	$\mathbf{P}(\ X_{t+1} \mid X_t, Y_t)$
$+x_t$	$+y_t$	$+x_{t+1}$	0.8
$+x_t$	$+y_t$	$-x_{t+1}$	0.2
$+x_t$	$-y_t$	$+x_{t+1}$	0.5
$+x_t$	$-y_t$	$-x_{t+1}$	0.5
$-x_t$	$+y_t$	$+x_{t+1}$	0.6
$-x_t$	$+y_t$	$-x_{t+1}$	0.4
$-x_t$	$-y_t$	$+x_{t+1}$	0.8
$-x_t$	$-y_t$	$-x_{t+1}$	0.2

Y_t	Y_{t+1}	$\mathbf{P}(\ Y_{t+1} \mid Y_t)$
$+y_t$	$+y_{t+1}$	0.6
$+y_t$	$-y_{t+1}$	0.4
$-y_t$	$+y_{t+1}$	0.2
$-y_t$	$-y_{t+1}$	0.8

You observe the evidence up to $t = 2$ as $(+e_1, -f_1, -e_2, +f_2)$ and want to infer $P(X_2, Y_2 | E_1 = +e_1, F_1 = -f_1, E_2 = -e_2, F_2 = +f_2)$.

Throughout this problem, you may answer as either numeric expressions (e.g. 0.03+0.1∗0.5) or numeric values (e.g. 0.08), or **None** if you think no result can be obtained based on given information.

(a) [1 pt] *Prior Sampling.* The following five samples were generated from prior sampling. What is the sample based estimate of $P(Y_2 = +y_2 \mid E_1 = +e_1, F_1 = -f_1, E_2 = -e_2, F_2 = +f_2)$?

$-x_1$	$-y_1$	$+x_2$	$+y_2$	$+e_1$	$-f_1$	$-e_2$	$+f_2$
$-x_1$	$+y_1$	$+x_2$	$-y_2$	$+e_1$	$-f_1$	$-e_2$	$+f_2$
$+x_1$	$-y_1$	$+x_2$	$+y_2$	$+e_1$	$+f_1$	$-e_2$	$+f_2$
$+x_1$	$+y_1$	$-x_2$	$+y_2$	$+e_1$	$-f_1$	$-e_2$	$+f_2$
$+x_1$	$-y_1$	$-x_2$	$-y_2$	$+e_1$	$-f_1$	$-e_2$	$-f_2$

Answer: ____________________________

(b) [1 pt] *Rejection Sampling.* You generate samples of $(X_1, Y_1, X_2, Y_2, E_1, F_1, E_2, F_2)$, variable by variable. In the first sample, you have already sampled the first four variables as $-x_1, -y_1, +x_2, +y_2$ and have not yet sampled (E_1, F_1, E_2, F_2). What is the probability of this sample being rejected?

Answer: ____________________________

(c) *Likelihood Weighting.* The following two samples were generated with likelihood weighting.

$$\begin{array}{cccccccc} -x_1 & -y_1 & +x_2 & +y_2 & +e_1 & -f_1 & -e_2 & +f_2 \\ -x_1 & -y_1 & +x_2 & -y_2 & +e_1 & -f_1 & -e_2 & +f_2 \end{array}$$

(i) [1 pt] What is the weight of the first sample?

Answer: ________________________

(ii) [1 pt] What is the weight of the second sample?

Answer: ________________________

(iii) [1 pt] What is the sample-based estimate of $P(Y_2 = +y_2 \mid E_1 = +e_1, F_1 = -f_1, E_2 = -e_2, F_2 = +f_2)$?

Answer: ________________________

(d) [2 pts] *Gibbs Sampling.* You want to use Gibbs Sampling to estimate $P(Y_2 = +y_2 \mid E_1 = +e_1, F_1 = -f_1, E_2 = -e_2, F_2 = +f_2)$, choosing to ignore evidence at $t = 3$ and onward.
The current sample is

$$-x_1 \quad -y_1 \quad +x_2 \quad +y_2 \quad +e_1 \quad -f_1 \quad -e_2 \quad +f_2$$

and the next step is to resample X_1, what is the probability that the new assignment to X_1 is $+x_1$?

Answer: ________________________

(e) *Particle Filtering.* You want to estimate $P(X_2, Y_2 | E_1 = +e_1, F_1 = -f_1, E_2 = -e_2, F_2 = +f_2)$ using particle filtering.

(i) [1 pt] At $t = 1$, you have a single particle $(X_1 = -x_1, Y_1 = -y_1)$. After passing it through the transition model, what is the probability of this particle becoming $(X_2 = +x_2, Y_2 = +y_2)$?

Answer: ________________________

(ii) [1 pt] Suppose after passing the sample through the transition model, you have the particle: $(X_2 = +x_2, Y_2 = +y_2)$. What is the weight of this particle after the observe update?

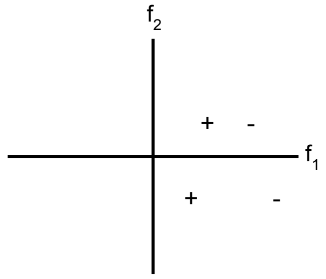
Answer: ________________________

(f) [2 pts] You now want to estimate $P(X_t, Y_t | e_{1:t}, f_{1:t})$ for some large t. You have limited computational resources and can either get N samples before rejection for rejection sampling, get N samples with likelihood weighting, or track of N particles using particle filtering.
Rank these three algorithms, indicating "1" for the algorithm which will give the most accurate sample-based approximation for $P(X_t, Y_t | e_{1:t}, f_{1:t})$, and "3" for the least accurate.

Rejection Sampling:__________ Likelihood Weighting:__________ Particle Filtering:__________

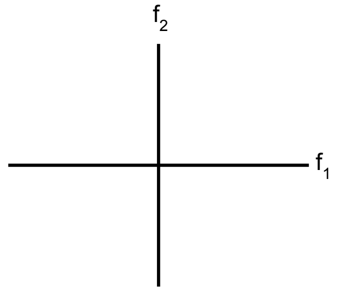
Q12. [15 pts] Decision Trees and Other Classifiers

(a) Suppose you have a small training data set of four points in *distinct* locations, two from the "+" class and two from the "–" class. For each of the following conditions, draw a particular training data set (**of exactly four points**: +, +, −, and −) that satisfy the conditions. If this is impossible, mark "Not possible". *If "Not possible" is marked, we will ignore any data points.*



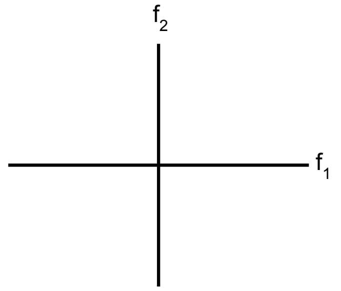
For example, if the conditions were "A depth-1 decision tree can perfectly classify the training data points," an acceptable answer would be the data points to the right.

(i) [2 pts] A linear perceptron with a bias term can perfectly classify the training data points, but a linear perceptron without a bias term cannot.



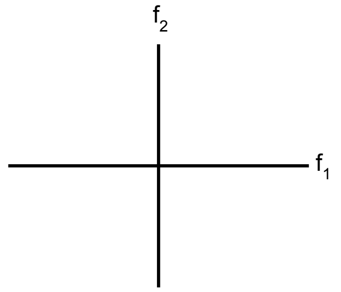
○ Not possible

(ii) [2 pts] A dual perceptron with a quadratic kernel function $K(x, z) = (1 + x \cdot z)^2$ can perfectly classify the training data points, but a linear perceptron with a bias term cannot.



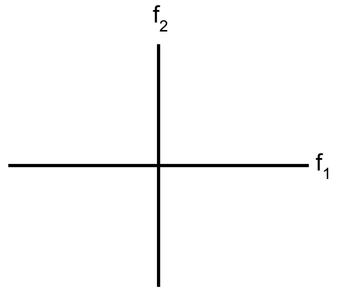
○ Not possible

(iii) [2 pts] A depth-2 decision tree can classify the training data points perfectly, but a dual perceptron with a quadratic kernel function $K(x, z) = (1 + x \cdot z)^2$ cannot.



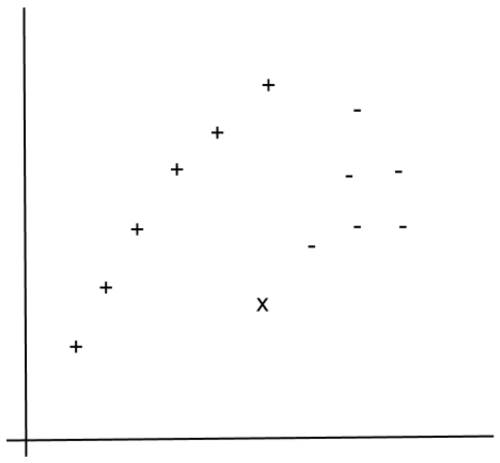
○ Not possible

(iv) [2 pts] A depth-2 decision tree cannot classify the training data perfectly



○ Not possible

(b) [2 pts] The plot below shows training instances for two classes ("+" and "-"). You use various methods to train a classifier between "+" and "-". You are then given a new point (marked by the "x") and use the previously trained systems to classify the new point.



Which of these methods are guaranteed to classify the new point as a "-"?

☐ Linear SVM (without using a bias term)
☐ Linear SVM (using a bias term)
☐ Dual perceptron with $K(x, z) = x \cdot z$
☐ Dual perceptron with $K(x, z) = x \cdot z + 1$
☐ None of the above

(c) [2 pts] Let $\{x_i, y_i | i = 1 \dots N\}$ be training examples and their class labels s.t. $y_i \in \{-1, 1\}$. Assume that the training data is separable when using a linear SVM with an additional bias feature.
For which of these kernel functions is the training data guaranteed to be separable using a dual perceptron?

☐ $K(x, z) = x \cdot z + 1$ ☐ $K(x, z) = x \cdot z$
☐ $K(x, z) = (x \cdot z + 1)^2$ ☐ None of the options

(d) You are still trying to classify between "+" and "-", but your two features now can take on only three possible values, $\{-1, 0, 1\}$. You would like to use a Naive Bayes model with the following CPTs:

X	$P(X)$
-	0.4
+	0.6

X	F_1	$P(F_1 \mid X)$
-	-1	0.4
-	0	0.5
-	1	0.1
+	-1	0.7
+	0	0.1
+	1	0.2

X	F_2	$P(F_2 \mid X)$
-	-1	0.1
-	0	0.1
-	1	0.8
+	-1	0.6
+	0	0.1
+	1	0.3

(i) [1 pt] If you observe that $F_1 = -1$ and $F_2 = -1$, how will you classify X using Naive Bayes?
○ $X = -$ ○ $X = +$

(ii) [1 pt] If you observe that $F_1 = 0$ and $F_2 = 0$, how will you classify X using Naive Bayes?
○ $X = -$ ○ $X = +$

(iii) [1 pt] If you observe that $F_1 = 1$ and $F_2 = 1$, how will you classify X using Naive Bayes?
○ $X = -$ ○ $X = +$