CS188 Summer 2025

Intro to Artificial Intelligence Midterm

Print Your Name: ______________________________

Print Your Student ID: ______________________________

Print Student name to your left: ______________________________

Print Student name to your right: ______________________________

You have 110 minutes. There are 7 questions of varying credit. (100 points total)

Question:	1	2	3	4	5	6	7	Total
Points:	19	16	11	16	12	13	13	100

For questions with **circular bubbles**, you may select only one choice.

- ○ Unselected option (Completely unfilled)
- ◎ Don't do this (it will be graded as incorrect)
- ● Only one selected option (completely filled)

For questions with **square checkboxes**, you may select one or more choices.

- ■ You can select
- ■ multiple squares
- ☑ Don't do this (it will be graded as incorrect)

Anything you write outside the answer boxes or you ~~cross out~~ will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation.

Read the honor code below and sign your name.

> By signing below, I affirm that all work on this exam is my own work. I have not referenced any disallowed materials, nor collaborated with anyone else on this exam. I understand that if I cheat on the exam, I may face the penalty of an "F" grade and a referral to the Center for Student Conduct.

Sign your name: ______________________________

Q1 *Probability: Chances Are Good* **(19 points)**

In this question, consider the below probability distributions over three variables: A (activity), V (visitor) and G (grade).

A	$P(A)$
sleep	0.3
study	0.4
party	0.3

V	$P(V)$
true	0.2
false	0.8

G	A	$P(G \mid A)$
pass	sleep	0.3
pass	study	0.9
pass	party	0.2
fail	sleep	0.7
fail	study	x
fail	party	0.8

G	A	V	$P(G \mid A, V)$
pass	sleep	true	0.2
pass	sleep	false	0.4
pass	study	true	0.75
pass	study	false	0.9
pass	party	true	0.2
pass	party	false	0.2
fail	sleep	true	0.8
fail	sleep	false	0.6
fail	study	true	0.25
fail	study	false	0.1
fail	party	true	0.8
fail	party	false	0.8

Q1.1 (2 points) What is the value of x in the $P(G \mid A)$ table?

Q1.2 (2 points) What is the value of $P(G = \text{pass} \mid A = \text{sleep})$?

Q1.3 (2 points) What is the value of $P(G = \text{pass}, A = \text{study})$?

Q1.4 (2 points) What is the value of $P(G = \text{pass} \mid A = \text{study}, V = \text{false})$?

Q1.5 (2 points) Select all expressions that can be used to compute $P(A, V, G)$, according to the chain rule.

- ☐ $P(A) \cdot P(V \mid A) \cdot P(G \mid A, V)$
- ☐ $P(G) \cdot P(A \mid V) \cdot P(G \mid A, V)$
- ☐ $P(V) \cdot P(A \mid V) \cdot P(G \mid V, A)$
- ☐ $P(G) \cdot P(A \mid G) \cdot P(V \mid G, A)$
- ☐ $P(G) \cdot P(G \mid A) \cdot P(G \mid A, V)$
- ○ None of the above

Q1.6 (2 points) What values of G and A make the below equation true? Select all that apply.

$$P(G \mid A,\ V = \text{true}) = P(G \mid A,\ V = \text{false})$$

- ☐ $G = \text{pass}, A = \text{sleep}$
- ☐ $G = \text{pass}, A = \text{party}$
- ☐ $G = \text{fail}, A = \text{study}$
- ☐ $G = \text{pass}, A = \text{study}$
- ☐ $G = \text{fail}, A = \text{sleep}$
- ☐ $G = \text{fail}, A = \text{party}$

Q1.7 (3 points) If $P(G = \text{fail}) = 0.35$, what is the value of $P(A = \text{sleep} \mid G = \text{fail})$?

The last two subparts are independent of previous subparts.

Q1.8 (2 points) Select all true statements about normalizing a probability table like $P(X \mid Y)$.

- ☐ Normalization eliminates rows that don't match the evidence variables.
- ☐ Normalization can be done by dividing each value by the sum of all the values in the table.
- ☐ Normalization eliminates any rows that have probability 0.
- ☐ Normalization adjusts the probabilities so they sum to 1.
- ☐ Normalization sums out any hidden variables.
- ○ None of the above

Q1.9 (2 points) Consider a Gridworld MDP with N states. The agent's available actions are North, South, East, West.

If we run Q-learning on this MDP, how many Q-values do we need to store? Express your answer as a function of N.

Q2 *Search: Pacman GhostBuster* **(16 points)**

Consider a variant of the Pacman game from lecture, with a single Pacman and a single Ghost:
1. The Ghost starts at an open square (i.e. a square with no wall) and does not move.
2. Then, Pacman can take any number of actions to reach the Ghost's square.
3. Once Pacman reaches the Ghost's square, the Ghost immediately moves to another open square, and does not move.
4. Then, Pacman can again take any number of actions to reach the Ghost's new square.
5. Pacman wins when he reaches the Ghost's new square (i.e. when Pacman has chased down the ghost twice).

Q2.1 (1 point) What is the maximum branching factor for this search problem?

○ 1 ○ 2 ○ 4 ○ 8 ○ 16 ○ 32

Q2.2 (3 points) Pacman knows the Ghost's two squares, i.e. it is constant, well-known information like the wall locations or the grid size.

Assume that Pacman's location is already in the state space. Which of these additions result in valid (not necessarily the most efficient) ways to model the Ghost in the search problem? Select all that apply.

☐ Add a Boolean variable in the state space representation.
The goal test checks only Pacman's location.

☐ Add a Boolean variable in the state space representation.
The goal test checks only the Boolean variable and Pacman's location.

☐ Add two Boolean variables in the state space representation.
The goal test checks only the Boolean variables.

☐ Add three Boolean variables in the state space representation.
The goal test checks only the Boolean variables.

○ None of the above

For Q2.3 and Q2.4, assume **all actions cost 1**.

Q2.3 (3 points) Let B be the branching factor, and S be the depth of the shallowest goal state in the search tree. Assume a finite-length solution always exists for this search problem.

Which search algorithms will always run in time complexity of $O\left(B^S\right)$ for this search problem? Select all that apply.

☐ DFS tree search

☐ BFS tree search

☐ UCS tree search

☐ Greedy tree search with the zero heuristic

☐ A* tree search with the zero heuristic

○ None of the above

Notation for the rest of the question:

- M is the Manhattan distance from Pacman to the Ghost.
- E is the Euclidean distance from Pacman to the Ghost.

Q2.4 (3 points) Select all admissible heuristics for this problem. (All actions still cost 1.)

- ☐ $M + E$
- ☐ $\min(M, E)$
- ☐ $M - E$
- ☐ $\max(M, E)$
- ☐ $\frac{1}{2}(M + E)$
- ○ None of the above

For Q2.5 and Q2.6, assume **East and West actions cost 3, and North and South actions cost 6**.

Q2.5 (3 points) Select all true statements about this search problem.

- ☐ M is an admissible heuristic.
- ☐ $(M \times 3)$ is an admissible heuristic.
- ☐ $(M \times 6)$ is an admissible heuristic.
- ☐ BFS tree search finds the optimal solution (assuming one exists).
- ☐ UCS tree search finds the optimal solution (assuming one exists).
- ○ None of the above

Q2.6 (3 points) Notation for this subpart:

- H is the horizontal distance from Pacman to the Ghost.
- V be the vertical distance from Pacman to the ghost.
- Example: If Pacman is on square $(2, 5)$ and the Ghost is on square $(6, 8)$, then $H = 4$ and $V = 3$.

Select all admissible heuristics for this search problem.

- ☐ $(3H + 3V)$
- ☐ $\min(3H, 6V)$
- ☐ $(6H + 3V)$
- ☐ $\max(3H, 6V)$
- ☐ $(3H + 6V)$
- ○ None of the above

Q3 *CSPs: Greenhouse Reassignment* **(11 points)**

You want to assign six plants (the variables) to three greenhouse buildings (the values).

The three buildings (values) are:

- Tropical (`Trop`)
- Temperate (`Temp`)
- Arid (`Arid`)

The six plants (variables) are:

- Bamboo (`Bam`) can only live in `Trop` or `Temp`.
- Basil (`Bas`) can only live in `Temp`.
- Fern (`Fer`) can only live in `Trop` or `Temp`.
- Succulent (`Suc`) can live in `Trop`, `Temp`, or `Arid`.
- Pepper (`Pep`) can only live in `Trop` or `Temp`.
- Orchid (`Ori`) can only live in `Trop`.

The plants also have these six constraints:

- `Bam` cannot be in the same zone as `Bas`.
- `Bas` cannot be in the same zone as `Fer`.
- `Bas` cannot be in the same zone as `Suc`.
- `Fer` cannot be in the same zone as `Suc`.
- `Fer` cannot be in the same zone as `Ori`.
- `Pep` cannot be in the same zone as `Ori`.

Q3.1 (3 points) After applying unary constraints, select the values that **remain** in each domain.

Domain of `Bam`:	☐ `Trop`	☐ `Temp`	☐ `Arid`
Domain of `Bas`:	☐ `Trop`	☐ `Temp`	☐ `Arid`
Domain of `Fer`:	☐ `Trop`	☐ `Temp`	☐ `Arid`
Domain of `Suc`:	☐ `Trop`	☐ `Temp`	☐ `Arid`
Domain of `Pep`:	☐ `Trop`	☐ `Temp`	☐ `Arid`
Domain of `Ori`:	☐ `Trop`	☐ `Temp`	☐ `Arid`

For the rest of the question, each subpart is **independent**: Every subpart starts with only unary constraints applied, and filtering from one subpart does not affect other subparts.

Q3.2 (2 points) We assign `Bam = Temp` and perform forward checking.
What values **remain** in the domain of `Bas`?

○ {`Trop`} ○ {`Temp`} ○ {`Arid`} ○ { }

Q3.3 (2 points) We assign `Fer = Trop` and perform forward checking.
What values **remain** in the domain of `Suc`?

○ {`Trop`} ○ {`Temp`} ○ {`Temp`, `Arid`} ○ { }

Q3.4 (2 points) We make no assignments, and we enforce arc consistently on only the `Bam` → `Bas` arc.
What values **remain** in the domain of `Bam`?

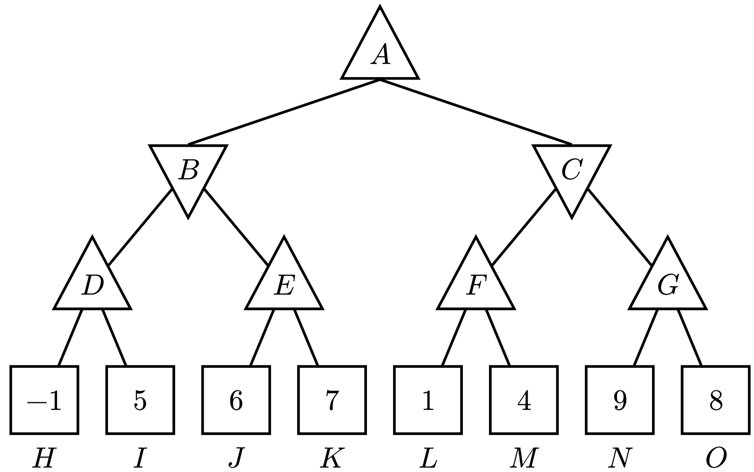
○ {`Trop`, `Temp`} ○ {`Temp`} ○ {`Trop`} ○ { }

Q3.5 (2 points) We make no assignments, and we enforce arc consistently on only the `Fer` → `Suc` arc.
What values **remain** in the domain of `Fer`?

○ {`Trop`} ○ {`Temp`} ○ {`Trop`, `Temp`} ○ { }

Q4 *Games* **(16 points)**

Consider the following minimax game tree:



For Q4.1 to Q4.4, assume no alpha-beta pruning takes place.

Q4.1 (1 point) What is the value at node A?

○ -1 ○ 5 ○ 6 ○ 7 ○ 1 ○ 4 ○ 9 ○ 8

Q4.2 (1 point) What is the value at node B?

○ -1 ○ 5 ○ 6 ○ 7 ○ 1 ○ 4 ○ 9 ○ 8

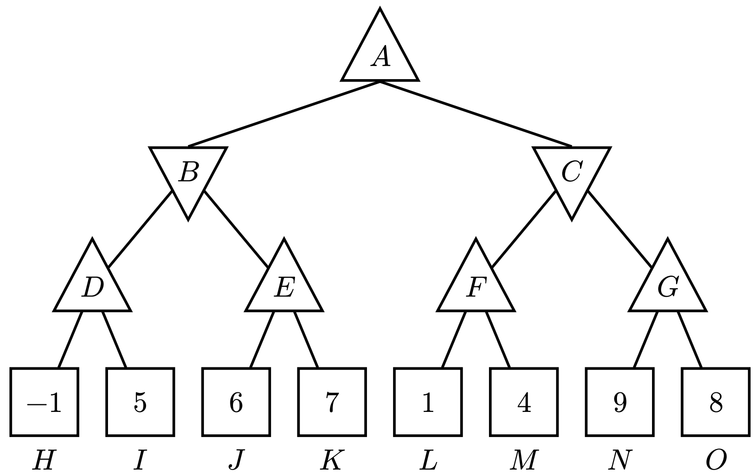
Q4.3 (1 point) What is the value at node C?

○ -1 ○ 5 ○ 6 ○ 7 ○ 1 ○ 4 ○ 9 ○ 8

Q4.4 (2 points) For this subpart only, B and C are changed from minimizer nodes to **chance nodes**. The children of the chance nodes all have equal probability.

What is the value at node A in this modified tree?

The game tree, reprinted for your convenience:



For the rest of the question, we run alpha-beta pruning, visiting nodes from left to right.

(Reminder: Alpha-beta pruning can cause some nodes to take on different values.)

Q4.5 (3 points) Which leaf nodes are **not visited** due to pruning? Select all that apply.

☐ H ☐ I ☐ J ☐ K ☐ L ☐ M ☐ N ☐ O

Q4.6 (2 points) What value does alpha-beta pruning assign to node D?

○ -1 ○ 5 ○ 6 ○ 7 ○ 1 ○ 4 ○ 9 ○ 8

Q4.7 (2 points) What value does alpha-beta pruning assign to node E?

○ -1 ○ 5 ○ 6 ○ 7 ○ 1 ○ 4 ○ 9 ○ 8

Q4.8 (2 points) What value does alpha-beta pruning assign to node A?

○ -1 ○ 5 ○ 6 ○ 7 ○ 1 ○ 4 ○ 9 ○ 8

Q4.9 (2 points) Suppose we change the value of node M from 4 to an unknown value x.

What values of x will cause node O **to be visited** when running alpha-beta pruning?

○ $1 < x < 8$ ○ $x > 5$ ○ $x < 1$

○ $x > 9$ ○ $5 < x < 9$ ○ $x < 5$

Q5 *MDPs: Clean Sweep* **(12 points)**

Alex is operating a robot vacuum cleaner that can be in one of three rooms: **A**, **B**, or **C**. The rooms are arranged in a line, as shown below.

Room **A**	Room **B**	Room **C**

$\leftarrow$ left right $\rightarrow$

At each timestep, the robot can take one of three actions:
- **Left**: Move to the room on the left. (If in **A**, stay in place.)
- **Right**: Move to the room on the right. (If in **C**, stay in place.)
- **Clean**: Attempt to clean the current room.

The Left and Right actions succeed with 100% probability.

The Clean action has three possible outcomes:
- The robot cleans the room and moves to the terminal state **success**.
- The robot tries to clean, but the dirt is too sticky, and it stays in the same room.
- The robot breaks and moves to the terminal state **broken**.

The probabilities for the Clean action are given in the table below:

s	$T(s,\ \text{Clean},\ \texttt{success})$	$T(s,\ \text{Clean},\ s)$	$T(s,\ \text{Clean},\ \texttt{broken})$
Room **A**	0.1	0.7	0.2
Room **B**	0.3	0.5	0.2
Room **C**	0.4	0.4	0.2

Unless otherwise specified, the discount factor is $\gamma = 1$.

Q5.1 (2 points) For this subpart only, consider this reward function:
- $+5$ for transitioning to **success**.
- -5 for transitioning to **broken**.
- 0 for all other transitions.

Alex wants the robot to go to a room and repeatedly take the Clean action until it ends up in a terminal state. Which room gives the highest expected return?

○ Room **A** ○ Room **B** ○ Room **C**

The table, reprinted for your convenience:

s	$T(s,\ \text{Clean},\ \texttt{success})$	$T(s,\ \text{Clean},\ s)$	$T(s,\ \text{Clean},\ \texttt{broken})$
Room **A**	0.1	0.7	0.2
Room **B**	0.3	0.5	0.2
Room **C**	0.4	0.4	0.2

For Q5.2 to Q5.4, consider this reward function:
- $+10$ for transitioning to **success**.
- -10 to transitioning to **broken**.
- -1 for all other transitions.

Q5.2 (6 points) Use value iteration to compute $V_0(s)$ and $V_1(s)$ for each state.

State s	$V_0(s)$	$V_1(s)$
Room **A**		
Room **B**		
Room **C**		

Q5.3 (2 points) Suppose the robot stays in one room and repeatedly takes the Clean action until it ends up in a terminal state. Which room gives the highest expected return?

○ Room **A** ○ Room **B** ○ Room **C**

Q5.4 (1 point) True or false: If γ is very small (e.g. $\gamma = 0.00000001$), the optimal policy for Rooms B and C will always be to clean immediately.

○ True ○ False

Q5.5 (1 point) For this subpart only, consider this reward function:
- $+50$ for transitioning to **success**.
- -50 for transitioning to **broken**.
- 0 for all other transitions.

True or false: After enough iterations of value iteration, the optimal policy will be to move to **C** and clean.

○ True ○ False

Q6 *RL: Sample Size Matters* **(13 points)**

Recall the TD learning update equation from lecture: $V^\pi(s) \leftarrow (1-\alpha)V^\pi(s) + (\alpha)[R + \gamma V^\pi(s')]$

Q6.1 (3 points) Which values come from the sample currently being processed? Select all that apply.

☐ s ☐ s' ☐ γ ☐ R

☐ $V^\pi(s)$ ☐ $V^\pi(s')$ ☐ α

Consider modifying the equation to process **4 samples** at a time (instead of 1 sample at a time).

- All 4 samples have the same starting state s.
- We want to update the $V^\pi(s)$ value using a **weighted average** of the return from the 4 samples. (Recall: "Return" includes both immediate reward and discounted future rewards.)
- The weights are four integers $1 \geq w_1 > w_2 > w_3 > w_4 \geq 0$, where $w_1 + w_2 + w_3 + w_4 = 1$.

Fill in the blanks for the modified equation:

$$V^\pi(s) \leftarrow (1-\alpha)V^\pi(s) + (\alpha) \underset{\text{(i)}}{\underline{\qquad}} \underset{\text{(ii)}}{\underline{\qquad}}$$

Q6.2 (2 points) Blank (i):

○ $\sum_{i=1}^{4}$ ○ $\prod_{i=1}^{4}$ ○ $\max_{i \in \{1,2,3,4\}}$ ○ $\frac{1}{4}\sum_{i=1}^{4}$

Q6.3 (2 points) Blank (ii):

○ $w_i R_i + \gamma V^\pi(s'_i)$ ○ $w_i(R_i + \gamma V^\pi(s'_i))$

○ $R_i + \gamma w_i V^\pi(s'_i)$ ○ $w_i(R_i + V^\pi(s'_i))$

Every time we use our modified update, we first sort the 4 samples from highest return to lowest.

Then, when applying the equation, we weight the highest-return sample with w_1, the second-highest-return sample with w_2, the third-highest-return sample with w_3, and the lowest-return sample with w_4.

Q6.4 (1 point) True or false: The highest-return sample always contributes more to the weighted average than the other samples.

○ True ○ False

Q6.5 (1 point) True or false: This modified update generally produces $V^\pi(s)$ estimates that are higher than the true values.

○ True ○ False

Q6.6 (4 points) Suppose that in every batch of four samples, you want to use only the highest-return sample in the TD learning update. You want to ignore the other three samples in the update.

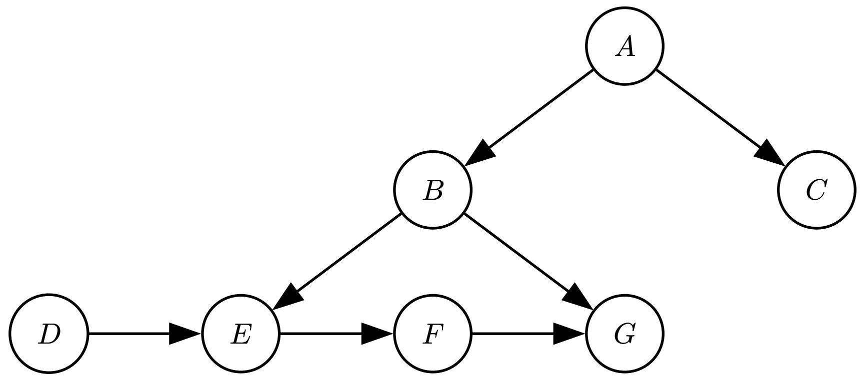
What values should you assign to the weights?

Remember that $1 \geq w_1 > w_2 > w_3 > w_4 \geq 0$, and $w_1 + w_2 + w_3 + w_4 = 1$.

$w_1 =$	$w_2 =$	$w_3 =$	$w_4 =$

Q7 *Bayes Net* **(13 points)**

Consider the Bayes Net below. All random variables are binary (each variable has two possible values).



Q7.1 (2 points) How many entries are in A's CPT (conditional probability table)?

○ 2 ○ 4 ○ 8 ○ 16

Q7.2 (2 points) How many entries are in E's CPT?

○ 2 ○ 4 ○ 8 ○ 16

For Q7.3 to Q7.6, use d-separation to determine if the independence assumption is true or false.

Q7.3 (1 point) $B \perp\!\!\!\perp C$

○ True (independent) ○ False (not independent)

Q7.4 (1 point) $B \perp\!\!\!\perp D \mid E$

○ True (independent) ○ False (not independent)

Q7.5 (2 points) $D \perp\!\!\!\perp G \mid F$

○ True (independent) ○ False (not independent)

Q7.6 (2 points) $B \perp\!\!\!\perp D \mid G$

○ True (independent) ○ False (not independent)

Q7.7 (3 points) $A \perp\!\!\!\perp D \mid$ ______

What variable can go into the blank line above so that the statement is **false**? In other words, what variable observation will make A and D **not** conditionally independent? Observing any common descendant of B and D will make them not independent.

Consider each choice separately (i.e. only one variable can go in the blank at a time). There may be multiple answers, so select all variables that work when used one at a time.

☐ B ☐ C ☐ E ☐ F ☐ G