CS188 Summer 2025

Intro to Artificial Intelligence Midterm

Solutions last updated: Sunday, July 27, 2025

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?

0.1

Solution:

$P(G = \text{pass} \mid A = \text{study}) + P(G = \text{fail} \mid A = \text{study}) = 1$

Since $P(G = \text{pass} \mid A = \text{study}) = 0.9$ from the table, we can derive the missing value.

$x = P(G = \text{fail} \mid A = \text{study}) = 0.1$

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

0.3

Solution: This value can be directly read off the table.

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

0.36

Solution: Use the product rule:

$$\begin{aligned} P(G = \text{pass}, A = \text{study}) &= P(A = \text{study}) \cdot P(G = \text{pass} \mid A = \text{study}) \\ &= 0.4 \cdot 0.9 \\ &= 0.36 \end{aligned}$$

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

0.9

Solution: This value can be directly read off the table.

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

☒ A $P(A) \cdot P(V \mid A) \cdot P(G \mid A, V)$

☒ B $P(V) \cdot P(A \mid V) \cdot P(G \mid V, A)$

□ C $P(G) \cdot P(G \mid A) \cdot P(G \mid A, V)$

□ D $P(G) \cdot P(A \mid V) \cdot P(G \mid A, V)$

☒ E $P(G) \cdot P(A \mid G) \cdot P(V \mid G, A)$

○ F None of the above

Solution:

(A), (B), and (E) are direct applications of the chain rule. They introduce the variables in different orders, but they all follow the chain rule formula.

(C) and (D) are incorrect because they introduce the variable G multiple times (i.e. multiple tables involving G get multiplied together). Also, they do not introduce all three variables per the chain rule.

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

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

□ A $G = \text{pass}, A = \text{sleep}$

□ B $G = \text{pass}, A = \text{study}$

☒ C $G = \text{pass}, A = \text{party}$

□ D $G = \text{fail}, A = \text{sleep}$

□ E $G = \text{fail}, A = \text{study}$

☒ F $G = \text{fail}, A = \text{party}$

Solution: This can be read directly off the table.

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

0.6

Solution: Use Bayes' rule:

$$\begin{aligned} P(A = \text{sleep} \mid G = \text{fail}) &= \frac{P(A = \text{sleep}) \cdot P(G = \text{fail} \mid A = \text{sleep})}{P(G = \text{fail})} \\ &= \frac{0.3 \cdot 0.7}{0.35} \\ &= 0.6 \end{aligned}$$

The last two subparts are independent of previous subparts.

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

☐ A Normalization eliminates rows that don't match the evidence variables.

☒ B Normalization can be done by dividing each value by the sum of all the values in the table.

☐ C Normalization eliminates any rows that have probability 0.

☒ D Normalization adjusts the probabilities so they sum to 1.

☐ E Normalization sums out any hidden variables.

○ F None of the above

Solution: Normalization is the process of making the sum of the probabilities in a distribution equal to 1 (D). This is done by dividing each probability by the total of all the probabilities (B). It does not filter probabilities (A, C). Nor does it marginalize (sum out variables) (E).

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.

$4N$

Solution: In Q-learning, we need to store one Q-value for every state-action pair.

There are 4 possible actions per state, and N states total. This gives a total of $4N$ state-action pairs.

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

Solution:

The maximum branching factor is determined by the maximum number of actions available to Pacman at any given time step.

Just like in the standard Pacman game, at any given timestep, Pacman can take some subset of the actions {North, South, East, West}, so the maximum branching factor is 4.

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.

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

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

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

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

Ⓔ None of the above

Solution:

(A) is false. Examining Pacman's location alone is not enough to determine whether the ghost has been chased down once or twice. For example, Pacman could be on the Ghost's second location, but could still be in the process of chasing down the ghost the first time.

(B) is true. We can use the Boolean variable to track whether Pacman has chased down the ghost the first time. Then, the goal test checks if the variable is `true` (i.e. the ghost has already been chased down once), and Pacman's location matches the Ghost's second location.

(C) is true. We can use the Boolean variables to track whether Pacman has chased down the ghost the first time and the second time. The goal test checks if both variables are `true`.

(D) is true. We know that two Boolean variables is enough, so adding an unnecessary third Boolean variable still results in a valid way to model the Ghost.

Note that in all choices, the successor function can be used to change the Boolean variable. For example, if Pacman takes an action that moves him into the Ghost's square, the successor function can output a successor state where the variable is now `true` instead of `false`.

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 $O(B^S)$ for this search problem? Select all that apply.

- ☐ A DFS tree search
- ☑ B BFS tree search
- ☑ C UCS tree search
- ☐ D Greedy tree search with the zero heuristic
- ☑ E A* tree search with the zero heuristic
- ○ F None of the above

Solution:

DFS tree search: False. In the worst case, DFS tree search could take $O(B^D)$ time, where D is the depth of the deepest goal state in the search tree.

BFS tree search: True. BFS tree search would find the optimal solution at depth S, exploring all $O(B^S)$ nodes with depth less than or equal to S.

UCS tree search: True. Since all actions cost 1, this behaves exactly like BFS.

Greedy tree search: False. With the zero heuristic, greedy search picks arbitrary successors off the queue, and in the worst case, it could behave like DFS tree search.

A* tree search: True. With the zero heuristic, A* tree search behaves like UCS tree search (which in turn behaves like BFS tree search).

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.)

☐ A $M + E$

☒ B $M - E$

☒ C $\frac{1}{2}(M + E)$

☒ D $\min(M, E)$

☒ E $\max(M, E)$

○ F None of the above

Solution: If there were no walls at all, the Manhattan and the Euclidean distances would each be a lower bound for the Pacman to go to the Ghost's location, and adding walls can only increase the true cost. Thus, M, E, or anything less (min, average, difference) must be admissible. The sum of the two could exceed the true cost (imagine the case with no walls).

Note that the rule about chasing down the ghost twice (instead of once) does not affect this answer.

If Pacman is currently in a state where he's chasing down the ghost for the first time, then the correct choices are still all underestimates, since the expression estimates the cost to catch the ghost the first time, and the cost to catch the ghost the second time is effectively estimated as 0.

If Pacman is currently in a state where he's chasing down the ghost for the second time, then the correct choices are still all underestimates, since the cost to goal is exactly the same as the cost to reach the ghost.

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.

☒ A M is an admissible heuristic.

☒ B $(M \times 3)$ is an admissible heuristic.

☐ C $(M \times 6)$ is an admissible heuristic.

☐ D BFS tree search finds the optimal solution (assuming one exists).

☒ E UCS tree search finds the optimal solution (assuming one exists).

○ F None of the above

Solution:

The Manhattan distance M will be less than cost of reaching goal even if no walls exist.

$M \times 3$ is also less because every action costs at least 3.

$M \times 6$ may exceed cost to goal - imagine the case where goal is to the left or right (those costs are 3).

BFS is optimal if action costs are 1, but not necessarily optimal if action costs are different.

UCS is optimal for any non-negative action costs.

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.

☑ A $(3H + 3V)$

□ B $(6H + 3V)$

☑ C $(3H + 6V)$

☑ D $\min(3H, 6V)$

☑ E $\max(3H, 6V)$

○ F None of the above

Solution: $3H + 6V$ is admissible, since it represents the relaxed scenario where there are no walls.

Any expression that is always less than $3H + 6V$ is also admissible.

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`:	☒ A `Trop`	☒ B `Temp`	☐ C `Arid`
Domain of `Bas`:	☐ A `Trop`	☒ B `Temp`	☐ C `Arid`
Domain of `Fer`:	☒ A `Trop`	☒ B `Temp`	☐ C `Arid`
Domain of `Suc`:	☒ A `Trop`	☒ B `Temp`	☒ C `Arid`
Domain of `Pep`:	☒ A `Trop`	☒ B `Temp`	☐ C `Arid`
Domain of `Ori`:	☒ A `Trop`	☐ B `Temp`	☐ C `Arid`

Solution: Answers come directly from the six plant variable bullet points above.

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`?

○ A {`Trop`} ○ B {`Temp`} ○ C {`Arid`} ● D { }

Solution: See solution after Q3.5.

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

○ A {`Trop`} ○ B {`Temp`} ● C {`Temp`, `Arid`} ○ D { }

Solution: See solution after Q3.5.

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`?

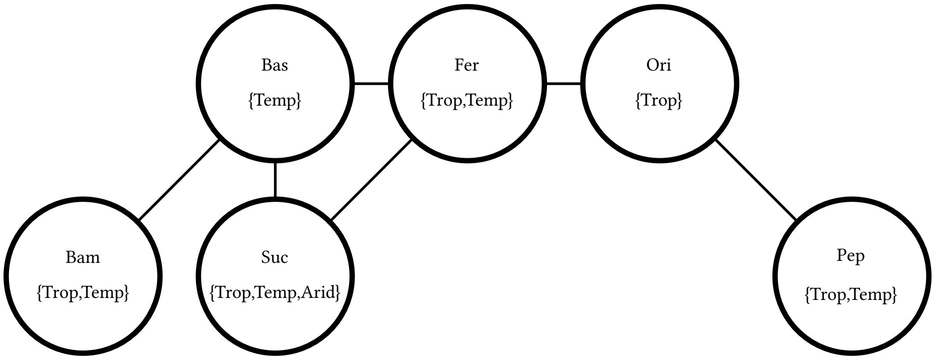
Ⓐ {`Trop`, `Temp`}   Ⓑ {`Temp`}   [✓ Ⓒ {`Trop`}]   Ⓓ { }

Solution: See solution after Q3.5.

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`}]   Ⓓ { }

Solution:



Consider the above CSP graph after enforcing unary constraints.

Q3.2: `Bas` domain before forward checking is `{Temp}`. Because of the constraint `Bam` ≠ `Bas`, we remove `Temp` from `Bas`'s domain after `Bam` is assigned as `{Temp}`.

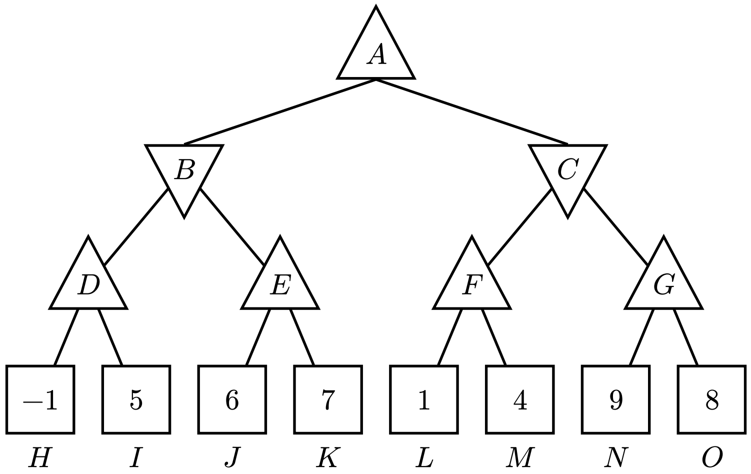
Q3.3: `Suc`'s domain before forward checking is `{Trop, Temp, Arid}`. The constraint `Fer` ≠ `Suc` disallows `Trop` for `Suc` once `Fer` is fixed to `{Trop}`. Removing `Trop` leaves `Suc`'s domain as `{Temp, Arid}`.

Q3.4: The tail is `Bam` with domain before arc consistency = `{Trop, Temp}`. We check all those values for constraint violations between `Bam` and `Bas`. The value `Temp` violates the constraint `Bam` ≠ `Bas`, so `Temp` is removed and `Bam`'s domain is now `{Trop}`.

Q3.5: The tail is `Fer` with domain before arc consistency = `{Trop, Temp}`. We check all those values for constraint violations between `Fer` and `Suc`. No values violate the constraint `Fer` ≠ `Suc`, so no values are removed and `Fer`'s domain remains `{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

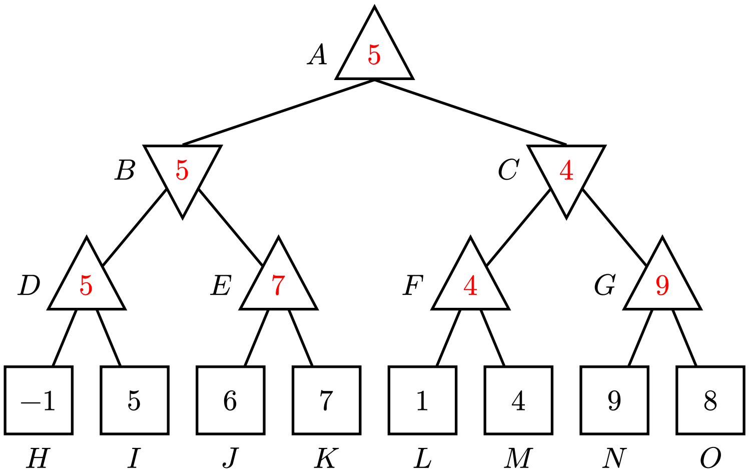
Q4.2 (1 point) What is the value at node B?

Ⓐ −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

Solution:

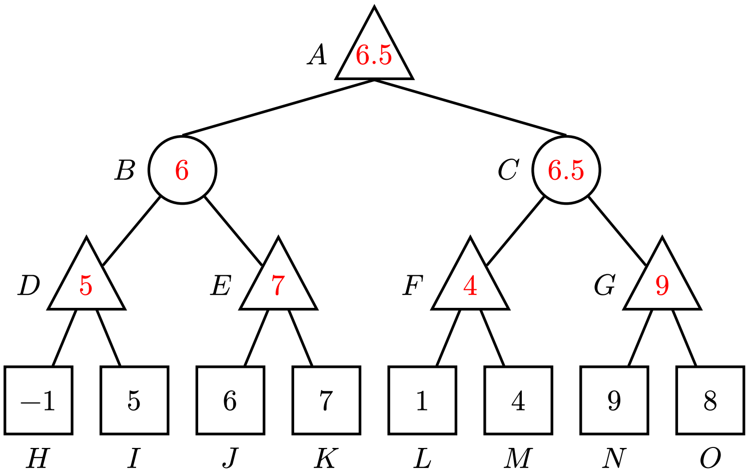


Q4.4 (2 points) For this subpart only, B and C are changed from minimizer nodes to **chance nodes**. The children of the chance nodes all have equal probability.

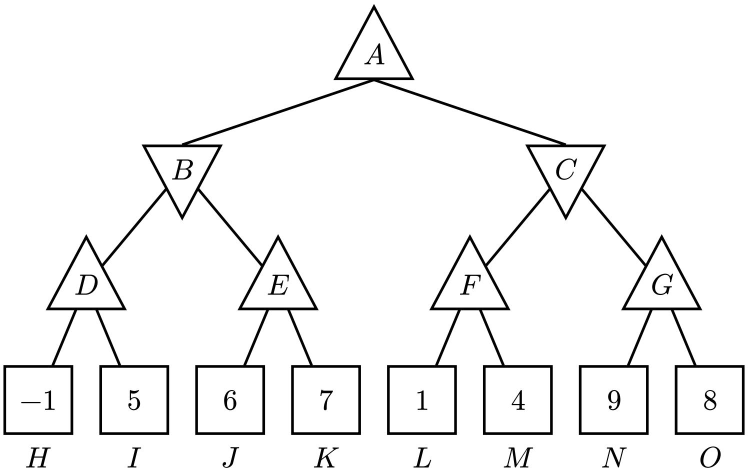
What is the value at node A in this modified tree?

6.5

Solution:



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.

A H B I C J [✓ D K]   E L F M [✓ G N]   [✓ H O]

(Question 4 continued...)

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

○ A -1 ● B 5 ○ C 6 ○ D 7 ○ E 1 ○ F 4 ○ G 9 ○ H 8

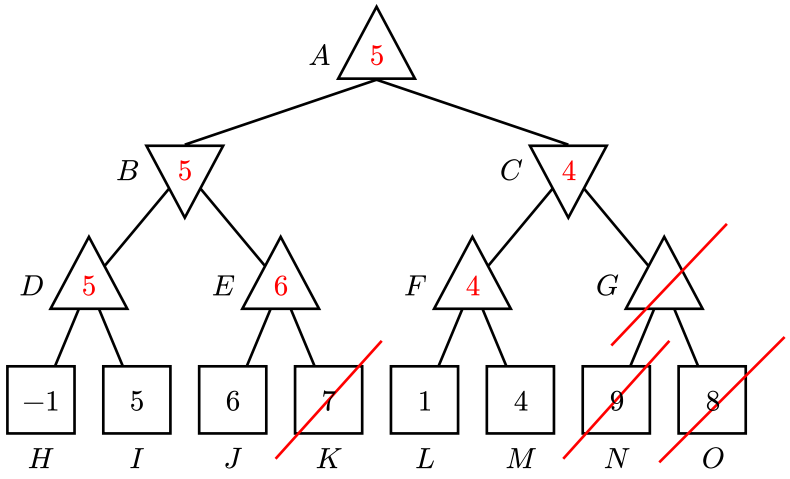
Q4.7 (2 points) What value does alpha-beta pruning assign to node E?

○ A -1 ○ B 5 ● C 6 ○ D 7 ○ E 1 ○ F 4 ○ G 9 ○ H 8

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

○ A -1 ● B 5 ○ C 6 ○ D 7 ○ E 1 ○ F 4 ○ G 9 ○ H 8

Solution:



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?

Ⓐ $1 < x < 8$

[✓ Ⓑ $x > 9$]

Ⓒ $x > 5$

Ⓓ $5 < x < 9$

Ⓔ $x < 1$

Ⓕ $x < 5$

Solution:

x must be greater than 5. Otherwise:

- If $x < 5$, then $F < 5$, and $C < 5$.
- Since $C < 5$ and can only get smaller (it's a min node), we know A will choose $B = 5$ over $C < 5$, and we can stop exploring C's descendants.
- Then O will not be visited.

x must also be greater than 9. Otherwise:

- If $x < 9$, then $F < 9$, and $C < 9$.
- At this point, A must choose between $B = 5$ and $C < 9$, and it's not clear yet which will be greater, so we must keep exploring.
- We then visit $N = 9$, which tells us $G \geq 9$.
- At this point, C must choose between $F < 9$ and $G > 9$.
- Since $G > 9$ and will only get bigger (it's a max node), we know C will choose $F < 9$ over $G > 9$, and we can stop exploring further.
- Then O will not get visited.

Combining these two conditions gives us $x > 9$.

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`
← left		right →

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`]

Solution: We compute the expected value $E[R_i]$ using:

$$E[R_i] = (p_{\text{success}} \cdot 5) + \left(p_{\text{stay}} \cdot E[R_i]\right) + (p_{\text{broken}} \cdot -5)$$

Solving:

$$\begin{aligned} E[R_{\text{A}}] &= -0.5 + (0.7 \cdot E[R_{\text{A}}]) && \to E[R_{\text{A}}] \approx -1.67 \\ E[R_{\text{B}}] &= 0.5 + (0.5 \cdot E[R_{\text{B}}]) && \to E[R_{\text{B}}] = 1.0 \\ E[R_{\text{C}}] &= 1.0 + (0.4 \cdot E[R_{\text{C}}]) && \to E[R_{\text{C}}] \approx 1.67 \end{aligned}$$

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**	**Solution:** 0	**Solution:** -1
Room **B**	**Solution:** 0	**Solution:** 0.5
Room **C**	**Solution:** 0	**Solution:** 1.6

Solution: For $V_0(s)$, all values are 0.

For $V_1(s)$, we compute the expected reward of the `clean` action for each room:

- Room **A**: $0.1 \cdot 10 + 0.7 \cdot (-1) + 0.2 \cdot (-10) = 1 - 0.7 - 2 = -1.7$
- Room **B**: $0.3 \cdot 10 + 0.5 \cdot (-1) + 0.2 \cdot (-10) = 3 - 0.5 - 2 = 0.5$
- Room **C**: $0.4 \cdot 10 + 0.4 \cdot (-1) + 0.2 \cdot (-10) = 4 - 0.4 - 2 = 1.6$

However, we also need to consider the value of taking a move action, which will always be -1. That has a greater reward for Room **A**, so that would be the action taken and the Value for that room.

So:
- $V_1(\text{A}) = -1$
- $V_1(\text{B}) = 0.5$
- $V_1(\text{C}) = 1.6$

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`]

Solution: Compute:

$$E[R_i] = (p_{\text{success}} \cdot 10) + \left(p_{\text{stay}} \cdot (E[R_i] - 1)\right) + (p_{\text{broken}} \cdot -10)$$

Solving:

- $E[R_{\text{A}}] \approx -5.67$
- $E[R_{\text{B}}] = 1$
- $E[R_{\text{C}}] \approx 2.67$

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

Solution: With small γ, only immediate rewards matter. Cleaning right away is best.

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

Solution: Room `C` has highest expected reward. There are no negative living rewards or discounts, so the optimal policy is to first move to `C`, then clean indefinitely.

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.

☑ A s

□ B $V^\pi(s)$

☑ C s'

□ D $V^\pi(s')$

□ E γ

□ F α

☑ G R

Solution:

The sample consists of a transition (s, a, s'), telling us that we observed a transition from state s, taking action a, and landing in state s'.

The sample also consists of a reward R that was obtained from this transition.

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):

● A $\sum_{i=1}^{4}$

○ B $\prod_{i=1}^{4}$

○ C $\max_{i\in\{1,2,3,4\}}$

○ D $\frac{1}{4}\sum_{i=1}^{4}$

Solution: To take a weighted average, we add up the four weighted samples.

Note that dividing by 4 is not necessary because the weights are already less than 1. For example, the weights might be $w_1 = 0.2, w_2 = 0.4, w_3 = 0.3, w_4 = 0.1$, and the returns might be $80, 60, 50, 100$. Then the weighted average is $0.2(80) + 0.4(60) + 0.3(50) + 0.1(100)$, and no dividing by 4 is necessary.

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

Ⓐ $w_i R_i + \gamma V^\pi(s_i')$ [✓ Ⓒ $w_i(R_i + \gamma V^\pi(s_i'))$]

Ⓑ $R_i + \gamma w_i V^\pi(s_i')$ Ⓓ $w_i(R_i + V^\pi(s_i'))$

Solution: The question asks for a weighted average of the return, which includes both immediate reward and discounted future rewards.

(A) is false because it does not weight future rewards. w_i is only applied on the immediate reward R_i, not the future rewards $\gamma V^\pi(s_i')$.

(B) is false because it does not weight the immedate reward. w_i is only applied on the future rewards $\gamma V^\pi(s_i')$, not the immediate reward R_i.

(C) is correct because it weights the entire return term, including immediate reward and discounted future rewards.

(D) is incorrect because it drops the γ discount factor, and the question asks for discounted future rewards.

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

Solution: True. The highest-return sample has the highest return, and it gets multiplied by the highest weight w_1, so the resulting product will be the largest product.

The other samples have lower returns and get multiplied by lower weights.

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

Solution: True. We merge the four samples into one using the weighted average, but our weighted average is biased toward the higher-return samples (which get multiplied by higher weights).

As a result, the merged sample return is higher than the true average of the four sample returns.

Using these higher values to update $V^\pi(s)$ will generally result in higher estimates.

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

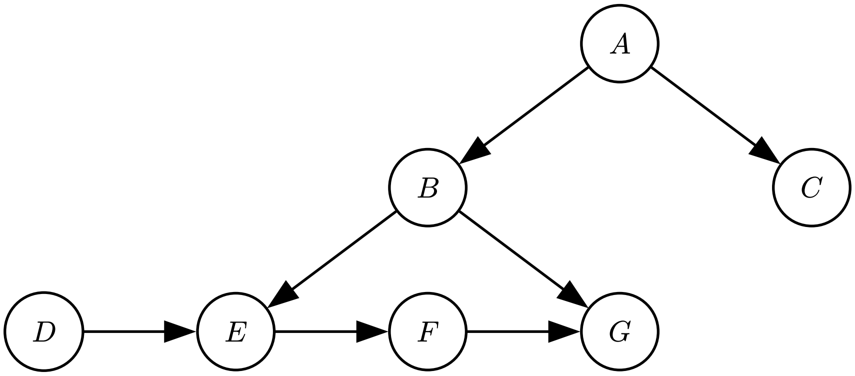
$w_1 = 1$	$w_2 = 0$	$w_3 = 0$	$w_4 = 0$

Solution: When you take a weighted average, weighting the highest-return sample by 1 and weighting the other samples by 0 results in the merged sample being identical to the highest-return sample.

For example, if you have sample returns $100, 80, 30, 5$, the weighted average is $1(100) + 0(80) + 0(30) + 0(5) = 100$, which is identical to the highest-return sample.

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

> **Solution:** A has no conditioning variable, thus the CPT can denote only possible values of $P(A)$. And each variable can have two values (binary), so the answer is 2.

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

Ⓐ 2   Ⓑ 4   [✓ Ⓒ 8]   Ⓓ 16

> **Solution:** E has two conditioning variables, B and D. Thus, the CPT should denote all possible values of $P(E \mid B, D)$, involving three variables where each has two possible values. Therefore, the answer is $2^3 = 8$.

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)]

> **Solution:** B and C has a common ancestor A, thus not independent.

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

Ⓐ True (independent)   [✓ Ⓑ False (not independent)]

> **Solution:** E is a common effect of B and D ("v-structure"), so observing E will make B and D not independent.

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

Ⓐ True (independent)   🅑 False (not independent)

Solution: As F is a common effect of D and B, thus the path D-E-B gets active. Together with the causal effect, B-G, the path D–E–B–G is active.

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

Ⓐ True (independent)   🅑 False (not independent)

Solution: The path B–E–D is active, because a descendant of E is observed.

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.

☐ A B ☐ B C ☑ C E ☑ D F ☑ E G

Solution: Observing E or its descendants F or G will all cause the $B - D - E$ common-effect triple to become active. This activates the $A - B - E - D$ path and causes A and D to become not independent.