CS 188 Fall 2024 — Introduction to Artificial Intelligence — Final Exam

Solutions last updated: Saturday, Dec 20, 2024

- You have 170 minutes.
- The exam is closed book, no calculator, and closed notes, other than two double-sided cheat sheets that you may reference.
- 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.

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

○ Unselected option (completely unfilled)

● Only one selected option (completely filled)

◎ Don't do this (it will be graded as incorrect)

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

■ You can select

■ multiple squares (completely filled)

☑ Don't do this (it will be graded as incorrect)

First name	
Last name	
SID	
Name of person to the right	
Name 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."

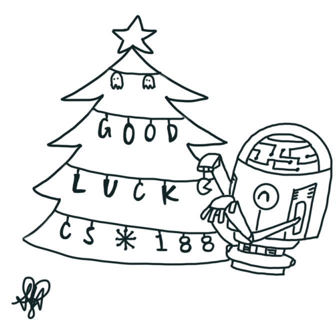
By signing below, I affirm that all work on this exam is my own work, and honestly reflects my own understanding of the course material. I have not referenced any outside materials (other than my cheat sheets), nor collaborated with any other human being on this exam. I understand that if the exam proctor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

Signature: ___________________________________

We hope you have a utility-maximizing holidays!

Point Distribution

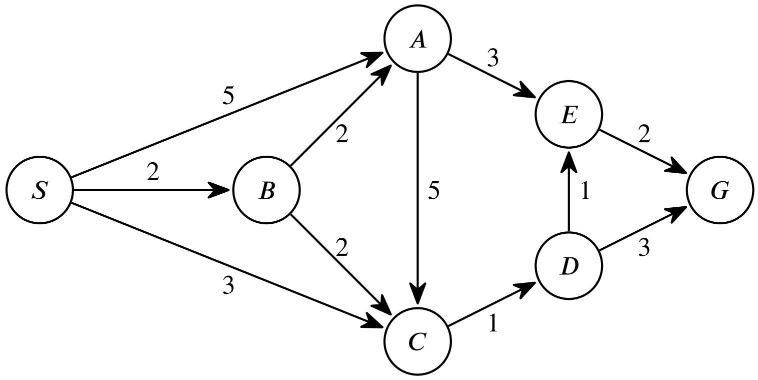
Q1.	Potpourri	9
Q2.	Acornomics	14
Q3.	Monotonic Alignment	18
Q4.	CSPeech	17
Q5.	Froogle Maps	13
Q6.	Machine Learning: Easter Island	18
Q7.	Square Bayes	11
	Total	100



Doodle credit: Samantha Huang

Q1. [9 pts] Potpourri

For the next three subparts, consider the search problem below, with start state S and a single goal state G.



Assume that when you expand a state, you add its successors on the fringe in alphabetical order. For example, if you expand B, then A is added to the fringe first, then C.

(a) [2 pts] What will be the path returned by DFS Graph Search on the search problem above?

Hint: DFS uses a stack. When expanding A, you push C, then you push E. This causes E to be popped off before C.

● $S \to C \to D \to G$
○ $S \to C \to D \to E \to G$
○ $S \to A \to C \to D \to E \to G$
○ $S \to A \to E \to G$

Pop S: {ABC}
Pop C: {ABD}
Pop D: {ABEG}
Pop G: end

(b) [1 pt] In BFS Graph Search, how many times does the successor function get called on state C?

○ 0 ● 1 ○ 2 ○ 3

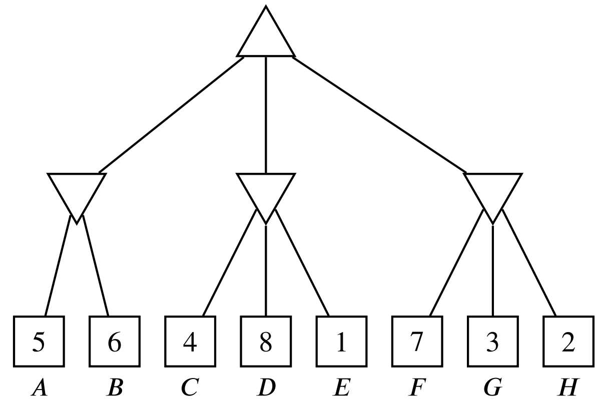
In BFS Graph Search, we only expand a state once, and after that, the state has been visited, so we never expand it again.

(c) [2 pts] In BFS Tree Search, how many times does the successor function get called on state C?

○ 0 ○ 1 ○ 2 ● 3

When we write ABC D, we mean that the fringe is now ABCD, with D newly added.
Pop Start: fringe is ABC.
Pop A: fringe is BCC E.
Pop B: fringe is CCE AC.
Pop C: fringe is CEAC D.
Pop C: fringe is EACD D.
Pop E: fringe is ACDD G.
Pop A: fringe is CDDG CE
Pop C: fringe is DDGCE D
Pop D: fringe is DGCE EG
Pop D: fringe is GCEEG EG
Pop G: end
Therefore, C is popped off 3 times.

For the next two subparts, consider the following minimax tree. The upwards arrows indicate maximizer nodes and downwards arrows indicate minimizer nodes. The squares indicate terminal leaf nodes.



(d) [1 pt] What is the value of the game tree at the root node?

○ 1 ○ 3 ● 5 ○ 7
○ 2 ○ 4 ○ 6 ○ 8

(e) [3 pts] We run alpha-beta pruning from left to right.
Select all nodes that will be pruned from the game tree.

☐ *A* ■ *D* ☐ *G*
☐ *B* ■ *E* ■ *H*
☐ *C* ☐ *F* ○ None

Q2. [14 pts] Acornomics

This clarification was given on the exam.

Q2(c), (d), and (e) only, the rightmost column should be:

☐ ~~$1+p$~~ → ☐ $1 - p\gamma$

☐ ~~$1+\gamma$~~ → ☐ $1 + p\gamma$

☐ ~~∞~~ → ☐ $\frac{1}{1-p\gamma}$

Consider a squirrel living in a tree. Each day, the squirrel has two possible actions:

- **Relax**: Stay in the tree. This always earns a reward of $r_{\text{relax}} > 0$.
- **Gather**: Attempt to gather an acorn.
  - With probability p, the squirrel succeeds and earns a reward of $r_{\text{acorn}} > 0$.
  - With probability $(1 - p)$, the squirrel is caught by a hawk and earns a reward of 0. Once the squirrel is caught, no further actions or rewards are possible.

In this question, consider two policies for the squirrel:

- **Always Relax** π_{relax}: The squirrel always chooses the Relax action.
- **Always Gather** π_{gather}: The squirrel always chooses the Gather action.

Hint: Recall the sum of a geometric series: $\sum_{n=0}^{\infty} a_1 \cdot r^n = \dfrac{a_1}{1-r}$.

For each subpart, select the **minimal** set of terms, such that their product corresponds to the answer. For example, if you think the answer is $\dfrac{p}{1+p}$, then select **only** p **and** $\dfrac{1}{1+p}$.

For the next two subparts, assume an infinite horizon with **no discounting**.

(a) [2 pts] What is $V^{\pi_{\text{relax}}}$, the expected return if the squirrel follows the Always Relax policy?

☐ p ☐ $\frac{1}{p}$ ☐ r_{relax} ☐ $1+p$

☐ $1-p$ ☐ $\frac{1}{1-p}$ ☐ r_{acorn} ■ ∞

(b) [3 pts] What is $V^{\pi_{\text{gather}}}$, the expected return if the squirrel follows the Always Gather policy?

■ p ☐ $\frac{1}{p}$ ☐ r_{relax} ☐ $1+p$

☐ $1-p$ ■ $\frac{1}{1-p}$ ■ r_{acorn} ☐ ∞

With a conservative always-stay policy, since r_{relax} is positive and there is infinite timesteps and no discounting — the expected cumulative reward is infinite.

With a "greedy" go-for-the-acorns policy, we can write the following recursive equation:

$$V^{\pi_{\text{gather}}} = p[r_{\text{acorn}} + V^{\pi_{\text{gather}}}] + (1-p) \cdot 0$$

Then, by combining terms and solving for $V^{\pi_{\text{gather}}}$, we get the expression $\dfrac{p \cdot r_{\text{acorn}}}{1-p}$

For the next three subparts, assume an infinite horizon with a discount factor of γ, $0 < \gamma < 1$.

(c) [2 pts] What is $V^{\pi_{\text{relax}}}$, the expected return if the squirrel follows the Always Relax policy?

☐ γ ☐ $1-p$ ■ $\frac{1}{1-\gamma}$ ☐ $\frac{1}{1+p\gamma-\gamma}$ ☐ $1+p$

☐ $1-\gamma$ ☐ $1+p\gamma-\gamma$ ☐ $\frac{1}{p}$ ■ r_{relax} ☐ $1+\gamma$

☐ p ☐ $\frac{1}{\gamma}$ ☐ $\frac{1}{1-p}$ ☐ r_{acorn} ☐ ∞

(d) [4 pts] What is $V^{\pi_{\text{gather}}}$, the expected return if the squirrel follows the Always Gather policy?

☐ γ ☐ $1-p$ ☐ $\frac{1}{1-\gamma}$ ☐ $\frac{1}{1+p\gamma-\gamma}$ ☐ $1+p$

☐ $1-\gamma$ ☐ $1+p\gamma-\gamma$ ☐ $\frac{1}{p}$ ☐ r_{relax} ☐ $1+\gamma$

■ p ☐ $\frac{1}{\gamma}$ ☐ $\frac{1}{1-p}$ ■ r_{acorn} ■ ∞

During the exam, the option ∞ was replaced with $\frac{1}{1-p\gamma}$

(e) [3 pts] What value of r_{acorn} causes the squirrel's expected return to be the same in the Always Relax and Always Gather policies?

☐ γ ☐ $1-p$ ■ $\frac{1}{1-\gamma}$ ☐ $\frac{1}{1+p\gamma-\gamma}$ ■ $1+p$

☐ $1-\gamma$ ☐ $1+p\gamma-\gamma$ ■ $\frac{1}{p}$ ■ r_{relax} ☐ $1+\gamma$

☐ p ☐ $\frac{1}{\gamma}$ ☐ $\frac{1}{1-p}$ ☐ r_{acorn} ☐ ∞

During the exam, the option $1+p$ was replaced with $1-p\gamma$.

With a conservative always-stay policy, we have an expected return given by a geometric series: $V^{\pi_{\text{relax}}} = \frac{r_{\text{relax}}}{1-\gamma}$.

With a "greedy" go-for-the-acorns policy, we have a slightly more complex recursive equation:

$$V^{\pi_{\text{gather}}} = p[r_{\text{acorn}} + \gamma V^{\pi_{\text{gather}}}] + (1-p)\cdot 0$$

Then, by combining terms and solving for $V^{\pi_{\text{gather}}}$, we get the expression:

$$V^{\pi_{\text{gather}}} = \frac{p \cdot r_{\text{acorn}}}{1-p\gamma}$$

The question we want to answer is what is the threshold on r_{acorn} such that $V^{\pi_{\text{gather}}} = V^{\pi_{\text{relax}}}$? In other words, when is

$$\frac{r_{\text{relax}}}{1-\gamma} = \frac{p \cdot r_{\text{acorn}}}{1-p\gamma}?$$

We get the following: $r^*_{\text{acorn}} = \dfrac{r_{\text{relax}} \cdot (1-p\gamma)}{(1-\gamma)\cdot p}$

Q3. [18 pts] Monotonic Alignment

During the exam this clarification was given:

Q3(c) Assume that log 0 = - (negative infinity)

Consider the following situation:

- You have a dataset of audio–text pairs. Each data point consists of an audio sequence, and a corresponding text sequence indicating what word was spoken.
- Each text sequence is N letters long: $\{x_1, \ldots, x_N\}$.
- Each audio sequence is a sequence of T audio chunks: $\{a_1, \ldots, a_T\}$.
- In this question, an audio chunk is written as two capital letters, e.g. "HH," "AH," or "OW."
- For each audio–text pair, $T > N$.

Given an audio–text pair, you wish to perform **monotonic alignment** on the pair: Find a monotonically-increasing sequence of T indices that tells you, for each audio chunk, which letter in the word was said during that audio chunk. The first audio chunk must correspond to the first letter, and the last audio chunk must correspond to the last letter.

For example, consider this audio–text pair:

Audio: {HH, HH, AH, AH, AH, LL, LL, LL, OW, OW, OW, OW}   Text: {h, e, l, l, o}

One plausible alignment for this audio–text pair is {1 1 2 2 2 3 4 4 5 5 5 5}; an unlikely alignment is {1 2 3 4 5 5 5 5 5 5 5 5}.

To figure out which alignment is best, we are given an alignment map $\mathcal{X}$, containing the probability that each audio chunk corresponds to a particular letter. An example of an alignment map for the above audio–text pair is shown below.

For part (a) only, ignore the bolds and underlines; they are explained later.

x_5	O	0.0	0.0	0.2	0.3	$\underline{0.2}$	$\underline{0.2}$	$\underline{0.2}$	$\underline{0.2}$	$\underline{\mathbf{0.4}}$	$\underline{\mathbf{0.7}}$	$\underline{\mathbf{0.8}}$	$\underline{\mathbf{1.0}}$
x_4	L	0.0	0.0	0.1	$\underline{0.1}$	0.2	0.3	**0.3**	**0.4**	0.2	0.1	0.1	0.0
x_3	L	0.0	0.0	$\underline{0.1}$	0.1	0.2	**0.3**	0.4	0.2	0.2	0.1	0.1	0.0
x_2	E	0.0	$\underline{0.1}$	**0.5**	**0.5**	**0.4**	0.1	0.1	0.1	0.2	0.1	0.0	0.0
x_1	H	$\underline{\mathbf{1.0}}$	$\underline{\mathbf{0.9}}$	0.1	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0
i		HH	HH	AH	AH	AH	LL	LL	LL	OW	OW	OW	OW
	t	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}

(a) [2 pts] In the alignment map above, what does entry (t, i) in $\mathcal{X}$ represent?

Hint: Consider whether each option is a valid probability distribution.

● $P(x_i \mid a_t)$ ○ $P(x_i)$ ○ $P(a_t \mid x_{1:i})$ ○ $P(a_t \mid x_i)$

Option 1 is correct as the probabilities are normalized in each column (sums to 1), therefore the conditioned variable must be the audio chunk. While Option 2 would be normalized over the columns, the column-wise probabilities in $\mathcal{X}$ are different, therefore it cannot be the correct answer. Options 3 and 4 would be normalized row-wise.

For the next five subparts, we will model monotonic alignment as a search problem:

- **State space:** The entries in the map ($N \times T$ states in total).
- **Starting state:** The bottom-left entry in the map (i.e. audio chunk 1 corresponding to the first letter)
- **Goal test:** Reach the top-right entry in the map (i.e. the last audio chunk T corresponding to the last letter).
- **Actions:** From each state, there are up to two actions available:

  (1) Move right one entry (i.e. output the same index again).

(2) Move right one entry and up one entry (i.e. output the next index).

The optimal solution is the monotonic alignment with the highest probability. Some possible solutions to the example alignment map above:

- The probability of the alignment {1 1 2 2 2 3 4 4 5 5 5 5} is computed by multiplying the **bold numbers** together.
- The probability of the alignment {1 2 3 4 5 5 5 5 5 5 5 5} is computed by multiplying the underlined numbers together.

(b) [1 pt] Which of the following search algorithms is guaranteed to find the optimal solution for the search problem described on the previous page?

☐ DFS ☐ BFS ● Neither.

(c) [2 pts] Which of the following functions, if applied to each map entry p, converts the entries to "costs," such that UCS Graph Search (with no modifications) is guaranteed to find the optimal solution?

Consider each function independently. Select all that apply.

☐ $\log p$ ☐ $\exp p$ ■ $-\log p$ ☐ $-p$ ○ None.

Negative logarithm is the only correct answer. Log probabilities are additive, and in the range $(-\infty, 0]$. Negating log probabilities ensures that the lower probabilities are larger, and all are in the range $[0, \infty)$.

(d) [3 pts] For this subpart only, consider a modified table where entries are non-negative costs, i.e. entry (i, t) corresponds to the cost of moving to that entry.

Can this modified problem be solved with UCS Tree Search?

● Yes, because we are guaranteed to have no cycles and the state graph is finite.
○ Yes, because UCS Tree Search is always able to terminate on finite state graphs.
○ No, because we could encounter cycles in the state graph.
○ No, because UCS Graph Search allows us to backtrack and UCS Tree Search does not.

(e) [3 pts] What is the number of states expanded by UCS Graph Search on the monotonic alignment problem? Express your answer as a tight upper-bound.

Reminder: A state is expanded when you call the successor function on that state.

○ $O\left(2^T\right)$ ○ $O\left(2^{TN}\right)$ ○ $O(T)$ ● $O(TN)$

In the worst case, every path must be searched. In this scenario, no more than $TN - (N - 1)$ states (letter, audio) pairs will be expanded. $TN - (N - 1) = \mathcal{O}(TN)$

(f) [2 pts] For this subpart only, consider the **general alignment** problem. It is the same as the monotonic alignment problem, except that the output does not need to be monotonically increasing (e.g. {1 1 1 1 4 2 2 3 4 5 5 5} is now a valid solution for the example alignment map).

For any given alignment map, is the optimal solution to the monotonic alignment problem the same as the optimal solution to the general alignment problem?

○ Yes ● No

In the rest of the question, consider a totally different formulation of the monotonic alignment problem, using HMMs (independent of all earlier subparts):

- The hidden state H_t represents what letter was said during the audio chunk at time t. For example, in the alignment {1 1 2 3}, $h_1 = 1$, $h_2 = 1$, $h_3 = 2$, and $h_4 = 3$.
- The evidence state E_t represents the audio chunk at time t.

(g) [3 pts] In order for this HMM to properly model the monotonic alignment problem, which of the following entries in the transition function must be set to 0? Select all that apply.

- [ ] $P(H_{t+1} = 3 \mid H_t = 2)$
- [x] $P(H_{t+1} = 4 \mid H_t = 2)$
- [ ] $P(H_{t+1} = 3 \mid H_t = 3)$
- [x] $P(H_{t+1} = 5 \mid H_t = 1)$
- [x] $P(H_{t+1} = 2 \mid H_t = 3)$
- [x] $P(H_{t+1} = 3 \mid H_t = 5)$
- ◯ None of these.

The monotonic alignment problem assumes that the sequences are monotonically increasing. For example, if $H_t = 7$, then at the next time step, H_{t+1} can only be 7 or 8.

Therefore, any situation where H_t is some value k, but H_{t+1} is not k or $k + 1$, must be impossible (i.e. occur with probability 0).

(h) [2 pts] Select all the true assumptions in this HMM.

- [ ] All E_t are independent from each other.
- [x] $H_{1:t-1}$ is conditionally independent of $H_{t+1:T}$ given H_t.
- [ ] H_t is conditionally independent of all previous ($H_{1:t-2}$) and future ($H_{t+1:T}$) hidden states given H_{t-1}.
- [ ] E_t is conditionally independent of H_t given E_{t-1}.
- ◯ None of the above.

Q4. [17 pts] CSPeech

During the exam these clarifications were given.

Q4 subparts (e)-(i) are all independent from each other.
Q4 the sequence of 4 words "the", "???", "???", and "loudly" apply to the whole question

We are given the the first and last words of a four-word sentence. Our goal is to complete the sentence by predicting the second and third words.

the	???	???	loudly
(Position 1)	(Position 2)	(Position 3)	(Position 4)

In the first half of this question, consider using a CSP to solve the problem. The variables are Position 1 through Position 4 in the sentence, and the domains are the possible words. We treat Position 1 and Position 4 as already assigned.

Here are the words in the domains of each unassigned variable, and the type of each word.
(You don't need to know the definition of "article" or "adverb.")

Word	Type of Word
dogs	noun
cats	noun
swim	verb
bark	verb
meow	verb
the	article
loudly	adverb

This CSP has two constraints:

- The word at Position 2 must be a noun.
- If a word at Position i is a noun, then the word at Position $i + 1$ must be a verb.

(a) [1 pt] After enforcing unary constraints, what is the size of Position 2's domain?

○ 1 ● 2 ○ 3 ○ 4 ○ 5 ○ 6

There are 2 nouns.

(b) [2 pts] Continuing from the previous subpart: After enforcing arc consistency on only the Position 3 → Position 2 arc, what is the size of Position 3's domain?

○ 1 ○ 2 ● 3 ○ 4 ○ 5 ○ 6

All the non-verbs get removed from Position 3.

(c) [2 pts] You run backtracking search on the CSP above. Is the outputted solution guaranteed to be the best solution?

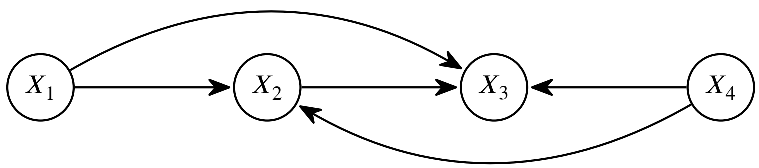
○ Yes, because backtracking search is complete and optimal.
○ Yes, because our constraints force backtracking search to output one specific solution.
● No, because backtracking search does not measure if one solution is "better" than another solution.
○ No, because backtracking search does not consider all solutions.

Option (D) is a true statement, because backtracking search returns after finding one solution. However, it is not the correct reason why the outputted solution is not best; even if backtracking search considered all solutions, it has no way of measuring which solution is better.

The rest of this question is independent from the earlier subparts.

In the second half of this question, consider using the Bayes Net below to solve this problem.

The Bayes Net has four nodes, corresponding to Position 1 (X_1) through Position 4 (X_4).



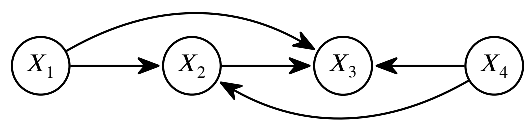
CPT (Conditional Probability Table) for X_3:

Row	X_3	X_2	$P(X_3 \mid X_2, x_1, x_4)$
1	dogs	dogs	0.1
2	swim	dogs	0.3
3	bark	dogs	0.6
4	dogs	swim	0.4
5	swim	swim	0.3
6	bark	swim	0.3
7	dogs	bark	0.4
8	swim	bark	0.3
9	bark	bark	0.3

CPT (Conditional Probability Table) for X_2:

Row	X_2	$P(X_2 \mid x_1, x_4)$
10	dogs	0.8
11	swim	0.1
12	bark	0.1

The Bayes Net, reprinted:



We want to find the values for X_2 and X_3 such that $P(x_1, x_2, x_3, x_4)$ is maximized.

(d) [1 pt] What variable(s) are observed as evidence in this problem? Select all that apply.

■ X_1 □ X_2 □ X_3 ■ X_4 ○ None

We know the values of x_1 and x_4, so we consider them observed evidence variables in our model.

(e) [3 pts] Recall that "dogs" is a noun, "swim" is a verb, and "bark" is a verb.

Pacman tells you that if a word at Position i is a noun, then the word at Position $i+1$ must be a verb.

Which row(s) in the Bayes Net should be set to 0 in order to correctly model Pacman's statement? Select all that apply.

■ Row 1 □ Row 4 □ Row 7 □ Row 10 ○ None
□ Row 2 □ Row 5 □ Row 8 □ Row 11
□ Row 3 □ Row 6 □ Row 9 □ Row 12

Row 1 is $P(\text{dogs}|\text{dogs})$, which should be 0 because this is an assignment where Position 2 is a noun and Position 3 is not a verb.

(f) [3 pts] Regardless of your answer to the previous subpart, suppose that we set only Row 5's value to 0.

Which row(s) in the Bayes Net should be updated such that the resulting tables are valid CPTs? Select all that apply.

☐ Row 1 ■ Row 4 ☐ Row 8 ☐ Row 11
☐ Row 2 ■ Row 6 ☐ Row 9 ☐ Row 12
☐ Row 3 ☐ Row 7 ☐ Row 10 ○ None

Rows 4, 5, and 6 form a valid probability distribution, so they should sum to 1.0.

If you zero out Row 5, then you need to adjust Rows 4 and 6 such that their values sum to 1.

(g) [2 pts] Suppose we use **prior sampling** to generate samples from this Bayes Net.

What is the probability of generating the sample "the dogs bark loudly"?

○ 0.32 ○ 0.60 ○ 1.00
○ 0.48 ○ 0.80 ● Not enough information.

We cannot compute the joint probability $P(x_1 = \text{the}, x_2 = \text{dogs}, x_3 = \text{bark}, x_4 = \text{loudly})$ because we do not know the probability of Word 1 being "the," and we do not know the probability of Word 4 being "loudly."

(h) [2 pts] Suppose we use **likelihood weighting** to generate samples from this Bayes Net.

What is the probability of generating the sample "the dogs bark loudly"?

○ 0.32 ○ 0.60 ○ 1.00
● 0.48 ○ 0.80 ○ Not enough information.

Probability of sampling "dogs" for Word 2 is 0.8.

Probability of sampling "bark" for Word 3, given that Word 2 was "dogs", is 0.6.

The total probability is $0.8 \times 0.6 = 0.48$.

(i) [1 pt] Suppose we use **likelihood weighting** to generate one sample from this Bayes Net.

True or false: The resulting sample (x_1, x_2, x_3, x_4) will always be the sample that maximizes $P(x_1, x_2, x_3, x_4)$.

○ True ● False

When we generate a sample from a Bayes Net, the resulting sample is not guaranteed to be the most likely assignment to the variables. We might get unlucky and generate a sample with lower probability in the joint distribution.

Q5. [13 pts] Froogle Maps

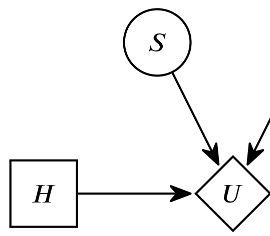
You own Froogle Maps, a navigation app. You model navigation as a search problem, where you know the state space and successor function (i.e. you know the map).

You have a choice of two heuristics: the Traffic heuristic, and the Distance heuristic. You choose exactly one heuristic to use.

After you choose a heuristic, you receive a query from a random user, who wants to know directions from a start state S to a goal state G. You know the probability distributions of S and G.

You run A* search with your chosen heuristic (H), the user's start state (S), and the user's goal state (G). The user is impatient: if you solve the problem within 10 seconds, you receive $U = 100$ utility points. Otherwise, you receive $U = 0$ utility points.

You wish to choose the heuristic that maximizes your expected utility. We can model this as a decision network:



(a) [2 pts] Which of the following is true about EU(Traffic) and EU(Distance), the expected utility of selecting each heuristic?

☐ If the Traffic heuristic dominates the Distance heuristic, then EU(Traffic) > EU(Distance).

☐ If the Distance heuristic always outputs 0, then EU(Distance) is always 0.

● Neither is true.

(A) is false. Suppose the Traffic heuristic takes a really long time to complete, and the Distance heuristic is very fast to compute. Then using the Traffic heuristic will always cause the query to time out, resulting in a utility of 0, while the Distance heuristic often finishes in time, resulting in a positive expected utility.

(B) is false. A trivial heuristic forces us to explore more states, but this could still be fast enough to return answers in time and achieve positive expected utility.

(b) [2 pts] For this subpart only, suppose the Traffic heuristic is inadmissible. Select all valid reasons for using an inadmissible heuristic.

■ We don't need the optimal solution.

■ The Traffic heuristic is easy to calculate.

○ Neither, we always want to use an admissible heuristic.

Fill in the blanks to describe how EU(Traffic) is calculated.

For each possible value of ___**(i)**___:

Step 1: Run A* search with the Traffic heuristic to see if a solution is found within 10 seconds.

Step 2: The utility is 100 if A* search finishes within 10 seconds, and 0 otherwise.

Step 3: Take this utility and multiply it by ___**(ii)**___.

Sum up every value you get in Step 3 above.

(c) [1 pt] Blank **(i)**:

● (S, G) ○ S ○ G ○ H

(d) [2 pts] Blank **(ii)**:

● $P(S, G)$ ○ $P(G)$ ○ 1

○ $P(S)$ ○ $U(S, G, H = \text{Traffic})$ ○ 0

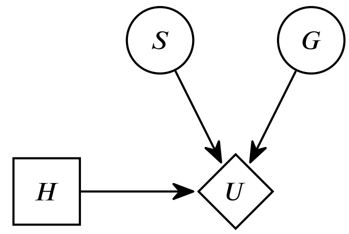
You want to consider whether Froogle Maps should partner with Big Brother, who will tell you the user's start state S.

For the next two subparts, refer to these tables, and the decision network from the previous page (reprinted below):

S	$P(S)$
Library	0.2
School	0.8

G	$P(G)$
Restaurant	0.6
Gym	0.4

S	G	H	$U(S,G,H)$
Library	Restaurant	Traffic	100
Library	Gym	Traffic	100
School	Restaurant	Traffic	0
School	Gym	Traffic	0
Library	Restaurant	Distance	100
Library	Gym	Distance	0
School	Restaurant	Distance	100
School	Gym	Distance	0



You would like to calculate the fair cost of Big Brother revealing S.

(e) [1 pt] If you know that the start state is Library, which is the better heuristic to use?

○ Distance ● Traffic

If we use the Distance heuristic, Library to Gym (occurs 40% of the time) would time out (utility 0), while Library to Restaurant (occurs 60% of the time) would finish in time (utility 100). Therefore, the expected utility of using Distance, given that S is library, is $0.4(0) + 0.6(100) = 60$.

If we use the Traffic heuristic, both Library to Gym and Library to Restaurant would finish in time (utility 100). Therefore, the expected utility of using Traffic, given that S is Library, is 100.

Since the expected utility of using Traffic is higher, the better heuristic to use is Traffic.

(f) [2 pts] Calculate MEU(S = School).

○ c0 ○ 40 ○ 80
○ 20 ● 60 ○ 100

EU(Traffic|S = School) $= 0.6(0) + 0.4(0) = 0$

If you know that the start state is School, then 60% of the time, the goal is Restaurant. With the Traffic heuristic, going from School to Restaurant achieves utility 0.

The other 40% of the time, the goal is Gym. With the Traffic heuristic, going from School to Gym achieves utility 0.

Therefore, the expected utility of using Traffic, given that the start state is School, is 0.

EU(Distance|S = School) $= 0.6(0) + 0.4(100) = 40$

If you know that the start state is School, then 60% of the time, the goal is Restaurant. With the Distance heuristic, going from School to Restaurant achieves utility 100.

The other 40% of the time, the goal is Gym. With the Distance heuristic, going from School to Gym achieves utility 0.

Therefore, the expected utility of using Distance, given that the start state is School, is 60.

The maximum expected utility is 40 (from using Distance), since this is greater than 0 (from using Traffic).

(g) [1 pt] Froogle Maps could also partner with Little Brother, who can tell you the user's goal state G.
True or False: VPI(S) + VPI(G) = VPI(S, G)

○ True ● False

$(VPI(S) + VPI(D) \neq VPI(S, D))$ VPIs are not additive.

(h) [2 pts] Big Brother offers to reveal the value of S, but only if Froogle Maps pays half of its **total** utility.

The inequality that represents whether you should accept this offer is: $\frac{1}{2}$ **(i)** $>$ **(ii)**

What expression goes in blank **(i)**?

- ○ VPI(S)
- ○ MEU($\varnothing$)
- ● MEU(S)
- ○ EU(Traffic | S = School)
- ○ EU(Traffic | S = Library)

What expression goes in blank **(ii)**?

- ○ VPI(S)
- ● MEU($\varnothing$)
- ○ MEU(S)
- ○ EU(Traffic | S = School)
- ○ EU(Traffic | S = Library)

If Froogle rejects Big Brother's offer, Froogle has utility equal to MEU($\varnothing$).

If Froogle takes Big Brother's offer, it will get utility equal to MEU(S). However, it then has to pay half of it to Big Brother; it only has utility of $\frac{1}{2}$ MEU(S).

Thus, we get the inequality $\frac{1}{2}$ MEU(S) $>$ MEU($\varnothing$). If the utility Froogle earns with Big Brother's information is better than the base utility, Froogle will take the offer.

Q6. [18 pts] Machine Learning: Easter Island

This clarification was given during the exam.

Q6(g) the answer choices in the left column should read w1=... and b1=... instead of w1=... and b2=...

The elves of Easter Island have left Petru the Paradise Dweller and are headed to the North Pole for their internship with Santa Claus—just in time for the holiday season! Help the elves solve challenges using machine learning.

The elves want to build a binary perceptron model to predict whether a gift will be enjoyed (-1 for Not Enjoyed, and $+1$ for Enjoyed). Each sample has two features: x_1, the Cost of the gift, and x_2, the Popularity of the gift.

Sample #	Cost (x_1)	Popularity (x_2)	Enjoyed (y)
#1	1	3	-1
#2	1	-1.5	-1
#3	2	2	$+1$
#4	3	1	-1
#5	4	1	$+1$

Note: For the next four subparts, **the first element in the weight vector is the bias term**.

(a) [3 pts] Using the Cost and Popularity features, which of the following weights $\mathbf{w}$ would allow you to linearly separate the samples above? Select all that apply.

- ☐ $\mathbf{w} = [4, 1, 1]$
- ☐ $\mathbf{w} = [2, -1, 1]$
- ☐ $\mathbf{w} = [-2, 1, -1]$
- ● None of the above.

(b) [3 pts] Which of the following additional features, if introduced, would result in the samples above being linearly separable? Consider each option independently. Select all that apply.

Note: You may assume that x_1, the Cost feature, always takes on integer values.

- ☐ $x_3 = |x_1| + |x_2|$
- ■ $x_3 = x_1 \cdot x_2$
- ■ $x_3 = (x_1 \bmod 2) \cdot |x_2|$
- ○ None of the above.

(c) [3 pts] In this subpart only, you introduce an additional feature, $x_3 = x_1 - x_2$. The initial weights are $\mathbf{w} = [1, 2, 2, 0]$. Perform a single round of weight updates using Sample #1. What are the new weights?

- ● $\mathbf{w} = [0, 1, -1, 2]$
- ○ $\mathbf{w} = [1, 2, 2, 0]$
- ○ $\mathbf{w} = [2, 3, 5, -2]$

(d) [1 pt] Suppose the elves introduce a new label, Indifferent. The elves re-label the dataset so that there are now 3 possible classes: Enjoyed, Not Enjoyed, and Indifferent. If we use a multi-class perceptron for this problem, what is the size of the weight matrix?

Note: We are using the Cost and Popularity features, not any additional x_3 feature.

- ○ $\mathbf{w} \in \mathbb{R}^{2\times2}$
- ● $\mathbf{w} \in \mathbb{R}^{3\times3}$
- ○ $\mathbf{w} \in \mathbb{R}^{4\times4}$
- ○ None of the above.

The following subparts are independent from the previous subparts.

A few of the elves are late to the start of their internship and need to be rescued from Svalbard, Norway. For Santa to pick them up, they need to get to the highest point of NewtonToppen, a mountain outside the city.

Consider the function $f(x) = -x^2 + 100$, where x is the elves' position, and $f(x)$ is the elevation at that position.

(e) [3 pts] The elves start at position $x_0 = 10$. They want to use the gradient ascent algorithm to reach the peak of the mountain (i.e. the position with maximum elevation). Their position update rule is: $x_{t+1} = x_t + \alpha(-2x_t)$, where α is the step size. Select all true statements.

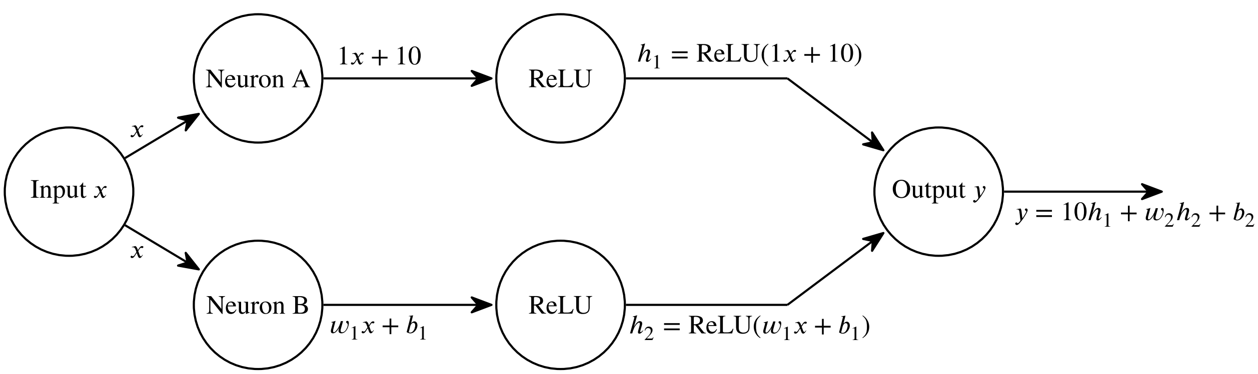
- ■ There exists a value of α such that the elves will reach the peak in fewer than 2 steps.
- ☐ If $\alpha < 0.5$, the elves will be at a negative position at some step t (i.e. $x_t < 0$ for some t).
- ☐ If $\alpha > 1$, gradient ascent will converge in the fewest steps possible.
- ○ None of the above.

As it turns out, Santa Claus doesn't trust the elves to reach the peak of NewtonToppen, so he plans to meet them at a point along the surface of the mountain. In order to land safely, Santa's autonomous sleigh needs to use a neural network to approximate the function above.

(f) [1 pt] True or False: To within any desired measure of accuracy $\epsilon > 0$, it is possible for a neural network (as defined in lecture) to model the function $f(x) = -x^2 + 100$.

● True ○ False

(g) [4 pts] Suppose that the neural network onboard Santa's sleigh has the given architecture:



The neurons perform a linear transformation. For example, Neuron A takes in x and outputs $1x + 10$.

The neural network is represented by the following equations, where all variables are scalars, and some of the weights have already been filled in for you:

$$h_1 = \text{ReLU}(1x + 10)$$
$$h_2 = \text{ReLU}(w_1 x + b_1)$$
$$y = 10h_1 + w_2 h_2 + b_2$$

Select values for the unknown weights w_1, b_1, w_2, and b_2, such that the resulting neural network gives the best approximation of the function $f(x) = -x^2 + 100$, for $-10 \le x \le 10$.

Select values for w_1 and b_1:

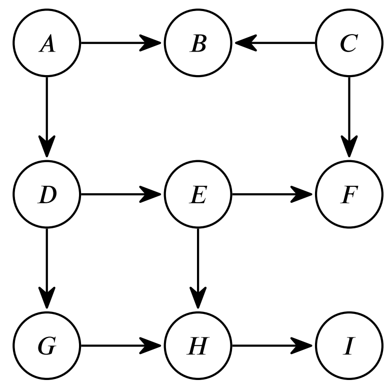
- ○ $w_1 = -1$, $b_2 = -10$
- ○ $w_1 = 2$, $b_2 = 10$
- ● $w_1 = 4$, $b_2 = 0$

Select values for w_2 and b_2:

- ○ $w_2 = 10$, $b_2 = 100$
- ● $w_2 = -5$, $b_2 = 0$
- ○ $w_2 = 0$, $b_2 = 25$

Q7. [11 pts] Square Bayes

For the first two subparts, consider the 9-node Bayes Net on the right:



(a) [2 pts] How many paths are there between A and H when considering if A and H are d-separated?

○ 1　　○ 2　　○ 3　　● 4

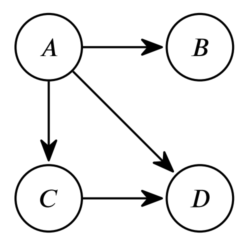
In total there are 4 paths total ABCFEH, ABCFEDGH, ADGH, and ADEH.

(b) [2 pts] Given the structure of the Bayes Net, which of the following are true? Select all that apply.

■ $A \perp\!\!\!\perp F \mid \{E\}$
■ $A \perp\!\!\!\perp I \mid \{D\}$
■ $B \perp\!\!\!\perp E \mid \{C, D, F, G, H\}$
☐ $D \perp\!\!\!\perp F \mid \{A, B, G, H\}$
○ None of the above.

1. true, b/c e d-separetes A from F
2. true, b/c d separates A from I
3. true, b/c markov blanket
4. false, b/c false because there is DEF path is causal

For the rest of the question, consider the 4-node Bayes Net to the right. Pacman wants to compute $P(D|B = b)$ using variable elimination.



(c) [2 pts] If Pacman eliminates A first, which factors should he join on?

■ $P(A)$　　☐ $P(D)$　　■ $P(D|A, C)$
☐ $P(b)$　　■ $P(b|A)$
☐ $P(C)$　　■ $P(C|A)$　　○ None

(d) [2 pts] What is the resulting factor after joining on A?

● $f(A, b, C, D)$　　○ $f(A, C, D)$　　○ $f(A, b, D)$　　○ $f(A, D)$
○ $f(b, C, D)$　　○ $f(A, b, C)$　　○ $f(A, C)$　　○ $P(A)$

(e) [1 pt] Blinky claims that it is more efficient (reduces the size of the largest factor generated) to join and eliminate C before A. Is Blinky correct?

● Yes　　○ No

After running variable elimination (not necessarily in the order above), the resulting remaining factors are $f(b)$ and $f(D)$.

Given these remaining factors, how should Pacman compute his desired query, $P(D \mid B = b)$?

$$P(D \mid B = b) = \frac{\textbf{(i)}}{\textbf{(ii)}}$$

(f) [1 pt] What goes in blank **(i)**?

- [x] $f(D)$
- [ ] $f(b)$
- [ ] $\sum_d f(d)$

(g) [1 pt] What goes in blank **(ii)**?

- [ ] $f(D)$
- [ ] $f(b)$
- [x] $\sum_d f(d)$