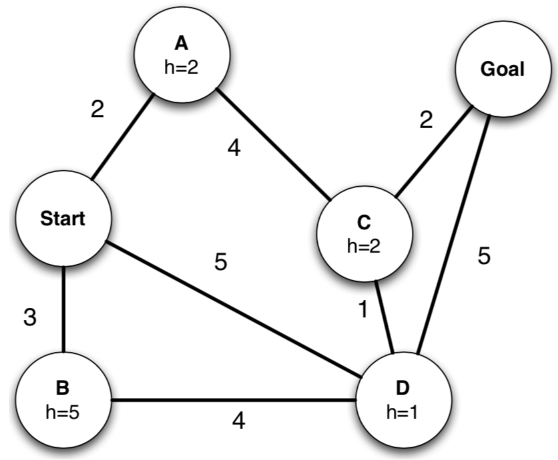
1 Big Picture

Problem	Representation	Stochastic/Deterministic	Goal
Search (all types)	States, Transitions, Start state, Goal state	Deterministic	Finding a path to the goal
CSPs	Domains, Variables, Constraints	Deterministic	Finding a complete assignment that satisfies constraints
LPs, IPs	Constraints, Objective function	Deterministic	Assignment that minimizes objective
Planning	Propositions, Actions, Start state, Goal state	Deterministic	Finding a plan (duh)
MDPs	States, Actions, Transition probabilities, Reward model	Stochastic	Finding a policy
Bayes Nets	CPT's, queries	Stochastic	Finding probabilities of specific events/queries
Game Theory	Choices, Payoffs	Deterministic	Finding a strategy that maximizes utility
Social Choice	Voting rules, candidates	Deterministic	Finding the winner

2 Search



For each of the following graph search strategies, work out the order in which states are explored, as well as the path returned by graph search. In all cases, break ties in alphabetical order. The start and goal state use letter S and G, respectively. Remember that in graph search, a state is explored only once.

(a) Depth-first search.

(b) Breadth-first search.

(c) Uniform cost search.

(d) Greedy search with the heuristic values h shown on the graph.

(e) $A^\star$ search with the same heuristic.

3 Adversarial Search

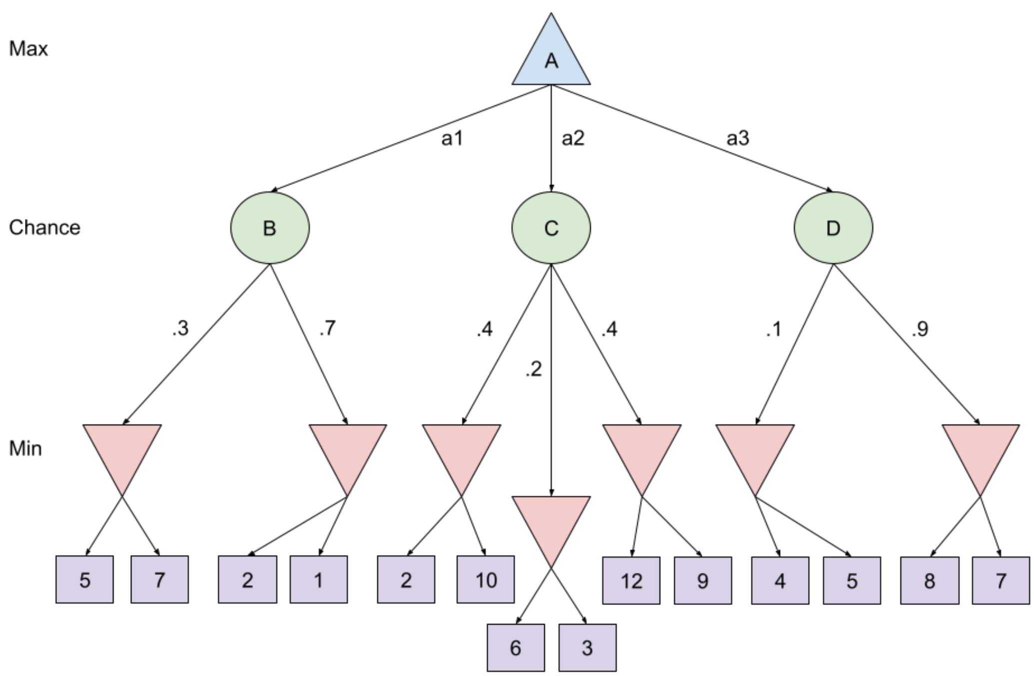
Warm up

1. What is the advantage of adding alpha-beta pruning to a minimax algorithm?

2. Give two advantages of Iterative Deepening minimax algorithms over Depth Limited minimax algorithms.

Expectiminimax

The following three questions are about the following adversarial "chance" tree.



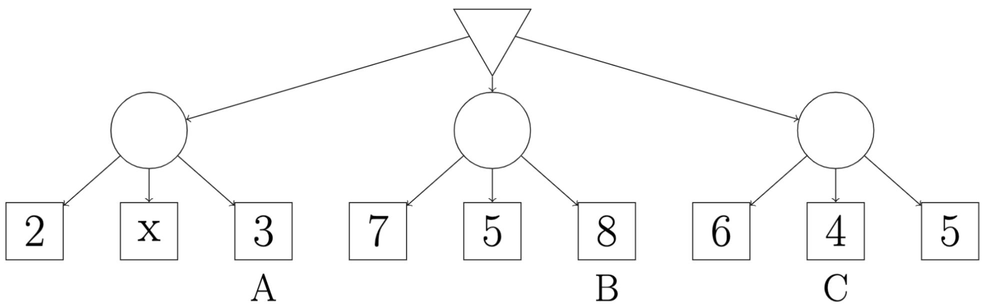
1. Calculate the EXPECTIMINIMAX values for nodes B, C and D in the above adversarial "chance" tree.

2. Which action will MAX choose, a_1, a_2, or a_3? Why?

3. If the utility values given for MIN were multiplied with a positive constant c, which action would MAX then choose?

Expectimin Pruning

For each of the leaves labeled A, B, and C in the expectimin tree below, determine which values of x would cause the leaf to be pruned, given the information that all values are non-negative and all nodes have 3 children. Assume all children of expectation nodes have equal probability and sibling nodes are visited left to right for all parts of this question. Assume we do not prune on equality.



Below, write your answers as one of (1) an inequality of x with a constant, (2) "none" if no values of x will cause the pruning, or (3) "any" if the node will be pruned for all values of x.

1. A:

2. B:

3. C:

4 CSP Backtracking Search

1. What is the difference between MRV and LCV?
2. What is the difference between forward checking and AC-3?
3. In this problem, you are given a 3×3 grid with some numbers filled in. The squares can only be filled with the numbers $\{2, 3, \dots, 10\}$, with each number being used once and only once. The grid must be filled such that adjacent squares (horizontally and vertically adjacent, but not diagonally) are relatively prime.

x_1	x_2	x_3
x_4	x_5	3
4	x_6	2

We will use backtracking search to solve the CSP with the following heuristics:

- Use the Minimal Remaining Values (MRV) heuristic when choosing which variable to assign next.
- Break ties with the Most Constraining Variable (MCV) heuristic.
- If there are still ties, break ties between variables x_i, x_j with $i < j$ by choosing x_i.
- Once a variable is chosen, assign the minimal value from the set of feasible values.
- For any variable x_i, a value v is infeasible if and only if: (i) v already appears elsewhere in the grid, or (ii) a variable in a neighboring square to x_i has been assigned a value u where $\gcd(v, u) > 1$, which is to say, they are not relatively prime.

Fill out the table below with the appropriate values.

- Give initial feasible values in set form; x_1 has already been filled out for you.
- Assignment order refers to the order in which the final value assignments are given. If x_i is the j^{th} variable on the path to the goal state, then the assignment order for x_i is j.
- In the branching column, write "yes" if the algorithm branches (considers more than one value) at that node in the search tree (for example, x_4 considers more than 1 value), and write "B" if the algorithm backtracks at that node, meaning it is the highest node in its subtree that fails for a value, and has to be chosen again. Also write the values it tried then failed.

Variable	Initial Feasible Values	Assignment Order	Final Value	Branch or Backtrack?
x_1	$\{5, 6, 7, 8, 9, 10\}$	________	______	__________
x_2	__________	________	______	__________
x_3	__________	________	______	__________
x_4	__________	________	______	__________
x_5	__________	________	______	__________
x_6	__________	________	______	__________

5 Local Search

1. Warm-up Questions

   (a) In what ways are genetic algorithms and local beam search similar? How are they different?

   (b) What is the difference between first-choice hill climbing and random-restart hill climbing?

   (c) What are some pros and cons of local beam search?

2. Of the local search algorithms we have discussed, which one(s) would perform best in a continuous state space and why?

3. What are the disadvantages and advantages of allowing sideways moves? How can we modify our search algorithm to address the disadvantages?

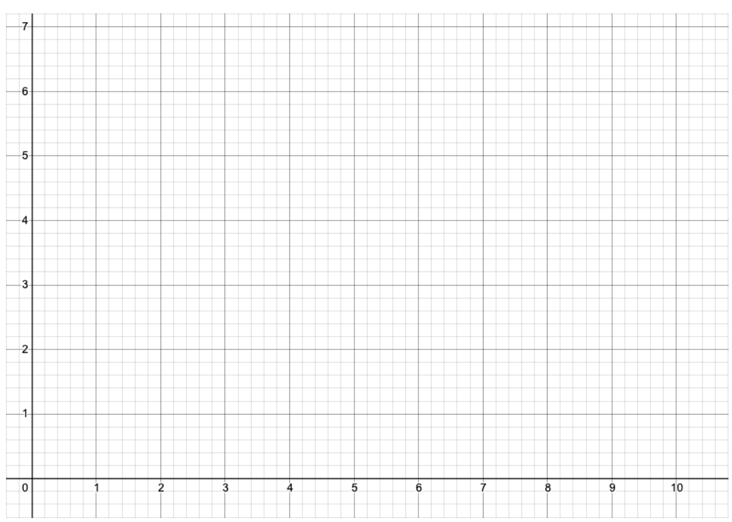
6 LP/IP

It's the holiday season and Santa Claus is getting ready to deliver lots of presents this year. However, he is struggling to pack everything, especially since there are a lot of perishable and fragile items. He needs help in determining how much dry ice and packing peanuts to use.

Let x_1 represent **pounds of dry ice** and let x_2 represent **pounds of packing peanuts**. We assume we can use a fraction of a pound of dry ice or packing peanuts. Santa has given us the following requirements:

- Santa only has room to spare for 10 bags of these materials. One pound of dry ice fits in one bag while one pound of packing peanuts takes up two bags.
- Santa needs to ensure that the temperature of the storage compartment of his sleigh is at most -2 degrees. One pound of dry ice **decreases** the temperature by 3 degrees while one pound of packing peanuts **increases** the temperature by 1 degree.
- Santa needs to ensure that the gifts are properly cushioned by these packaging materials. A pound of dry ice provides 2 units of "protectiveness" and a pound of packing peanuts provides 1 unit of "protectiveness". Santa wants there to be at least 3 units of "protectiveness".
- Unfortunately, Santa is on a tight budget this year so he is trying to minimize costs. One pound of dry ice costs \$0.5 and one pound of packing peanuts costs \$1.

1. Represent the following problem as an LP and graph the constraints in the provided graph.



2. Which rows in A correspond to the lines in the graph?

3. What would the optimal solution be?

4. If Santa didn't know the cost of dry ice and packing peanuts, what would be a cost vector that makes the optimal solution (1, 1)?

5. List four cost vectors that will lead to an infinite number of solutions.

6. If Santa can only add dry ice and packing peanuts in increments of 1 pound (ie. he cannot divide a pound up) which constraints (and corresponding priorities) get pushed onto the priority queue in the first iteration of Branch and Bound?

7. What is the IP solution to this problem?

7 Propositional Logic

1. Warm Up: Are you familiar with these terms?
   - Symbols
   - Operators
   - Sentences
   - Literals
   - Knowledge Base
   - Entailment
   - Query
   - Satisfiable
   - Valid
   - Clause - Definite, Horn clauses
   - Model Checking
   - Theorem Proving
   - Modus Ponens

2. Assume we have the following propositions: $BatteryDead, RadioWorks, OutOfGas$, and $CarStarts$ (You may use the abbreviations B, R, O, C in your answer).
   (a) What is the total number of possible models?
   (b) How many models are there in which the following sentence is $false$?
   $(RadioWorks \land CarStarts) \implies (\neg OutOfGas \land \neg BatteryDead)$
   (c) Is the above sentence equivalent to a set of Horn clauses? Explain.
   (d) Prove that the above sentence is *not* entailed by the sentence $RadioWorkds \implies \neg BatteryDead$.

3. From the knowledge base below, prove E.

A ∨ ¬D	A ∨ ¬B	C	D ∨ ¬C	¬A ∨ E ∨ ¬D	¬B

4. Indicate whether the following sentence is *valid*, *satisfiable*, or *unsatisfiable*. If satisfiable, give a model such that the sentence is satisfied. Prove your answer by reducing the sentence to its simplest form. Remember to **show all the steps and write down an explanation of each step.** Let T stand for the atomic sentence *True* and F for the atomic sentence *False*.
   $((T \Leftrightarrow \neg(x \lor \neg x)) \lor z) \land \neg(z \land ((z \land \neg z) \Rightarrow x))$

5. Indicate whether the following sentence is *valid*, *satisfiable*, or *unsatisfiable*. If satisfiable, give a model such that the sentence is satisfied. Prove your answer by reducing the sentence to its simplest form. Remember to **show all the steps and write down an explanation of each step.** Let T stand for the atomic sentence *True* and F for the atomic sentence *False*.

   $(\neg(x \vee \neg x) \wedge y) \vee ((x \vee (z \Rightarrow \neg z)) \wedge ((x \Rightarrow z) \vee \neg(F \Rightarrow T)))$

8 Satisfiability and Planning

Up until now we have assumed that the plans we create always make sure that an actions preconditions are satisfied. Let us now investigate what propositional successor-state axioms such as $HaveArrow^{t+1} \iff (HaveArrow^t \land \neg Shoot^t)$ have to say about actions whose preconditions are not satisfied.

(a) First, let us consider what successor-state axioms are. How do they differ from action axioms, and why might we choose to use them?

(b) Show that the axioms predict that nothing will happen when an action is executed in a state where its preconditions are not satisfied.

(c) Consider a plan p that contains the actions required to achieve a goal but also includes illegal actions. Is it the case that successor-state axiom will allow the actions?

We recommend that you write a truth table and ask yourself the following question when looking at the truth table:

- Can I shoot if I don't have an arrow?

Suppose we are tasked with making a plan to deliver N-95 masks around the U.S.
We use the following propositions below in a GraphPlan approach:

- $at(loc)$: our cargo plane is at location loc
- $fuel(x)$: the fuel level is at x , $x \in [0, 5]$.
- $hasFuel(loc)$: location loc has fuel (to re-fuel the plane with)
- $hasMasks(loc)$: location loc has masks

Our starting state is $at(Pittsburgh) \land fuel(5)$.

1. Define each action as an operator in the following table (note that we can drop the t parameter from each predicate and action):

	$refuel$	$fly(o, d)$	$deliver$
Precondition			
Add			
Delete			

2. Now draw the GraphPlan graph up to proposition level S_1. Suppose NYC is the only other location besides Pittsburgh.

3. Which operators are mutually exclusive in A_0? Which propositions are mutually exclusive in S_1?

4. In general, when does GraphPlan stop extending the graph?

5. Is GraphPlan sound? complete? optimal (with respect to the number of actions in the plan returned)?

9 Probability

1. Write all the possible chain rule expansions of the joint probability $P(a, b, c)$. No conditional independence assumptions are made.

2. Determine if the following statements are true or false. No independence assumptions are made.

   (a) $P(A, B) = P(A|B)P(A)$

   (b) $P(A|B)P(C|B) = P(A, C|B)$

   (c) $P(B, C) = \sum_{a \in A} P(B, C|A)$

   (d) $P(A, B, C, D) = P(C)P(D|C)P(A|C, D)P(B|A, C, D)$

3. Let A be a random variable representing the choice of protein in the sandwich with three possible values, $\{mutton, bacon, egg\}$, let B be a random variable representing the choice of bread with two possible values, $\{toast, naan\}$, and let K be a random variable representing the presence of ketchup or not, $\{+k, k\}$.

   How many values are in each of the probability tables and what do the entries sum to?
   Write **'?'** if there is not enough information given.

Table	**num**	**sum**
$P(A, B)$		
$P(A, B, +k)$		
$P(A, B \mid +k)$		
$P(B \mid +k, A)$		

4. Consider the following probability tables:

A	$P(A)$
+a	0.8
–a	0.2

A	B	$P(B\|A)$
+a	+b	0.9
+a	–b	0.1
–a	+b	0.6
–a	–b	0.4

B	C	$P(C\|B)$
+b	+c	0.8
+b	–c	0.2
–b	+c	0.8
–b	–c	0.2

C	D	$P(D\|C)$
+c	+d	0.25
+c	–d	0.75
–c	+d	0.5
–c	–d	0.5

   (a) Using the table above and the assumptions per subquestion, calculate the following probabilities given no independence assumptions. If it is impossible to calculate without more independence assumptions, specify the least number of independence assumptions that would allow you to answer the question (don't do any computation in this case).

(i) $P(+a, -b)$

(ii) $P(-a, -b, +c)$

(iii) Now assume C is independent of A given B and D is independent of everything else given C. Calculate $P(+a, -b, +c, +d)$ or say what other independence assumptions are necessary.

(b) Which of the following expressions indicate that X is independent of Y given Z?

(i) $P(X, Y|Z) = P(X|Z)P(Y|Z)$
(ii) $P(X|Y, Z) = P(X|Z)$
(iii) $P(X, Y, Z) = P(X, Z)P(Y)$
(iv) None of the above

(c) Which of the following expressions are equal to $P(R, S, T)$ given no independence assumptions?

(i) $P(R|S, T)P(S|T)P(T)$
(ii) $P(T, S|R)P(R)$
(iii) $P(T|R, S)P(R)P(S)$
(iv) $P(T|R, S)P(R, S)$
(v) $P(R|S)P(S|T)P(T)$
(vi) $P(R|S, T)P(S|R, T)P(T|R, S)$
(vii) None of the above

10 MDPs/RL

1. Warm Up

   - What does the Markov Property state?
   - What are the Bellman Equations, and when are they used?
   - What is a policy? What is an optimal policy?
   - How does the discount factor γ affect the agent's policy search? Why is it important?
   - What are the two steps to Policy Iteration?
   - What is the relationship between $V^*(s)$ and $Q(s, a)$?
   - Exploration, exploitation, and the difference between them? Why are they both useful?
   - What is the difference between on-policy and off-policy learning?
   - What is the difference between model-based and model-free learning?
   - We are given a pre-existing table of Q-values (and its corresponding policy), and asked to perform ϵ-greedy Q-learning. Individually, what effect does setting each of the following constants to 0 have on this process?

     (i) α:

     (iii) ϵ:

- For each of the following functions, write which MDP/RL value the function computes, or none if none apply. We are given an MDP (S, A, T, γ, R), where R is only a function of the current state s. We are also given an arbitrary policy π.

  Possible choices: $V^*, Q^*, \pi^*, V^\pi, Q^\pi$.

  (i) $f(s) = R(s) + \sum_{s'} \gamma T(s, \pi(s), s') f(s')$

  (ii) $g(s) = \max_a \sum_{s'} T(s, a, s')[R(s) + \gamma \max_{a'} Q^*(s', a')]$

2. MDPs - Micro-Blackjack: In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. Otherwise, you must Stop. When you Stop, your utility is equal to your total score (up to 5), or zero if you get a total of 6 or higher. When you Draw, you receive no utility. There is no discount ($\gamma = 1$).

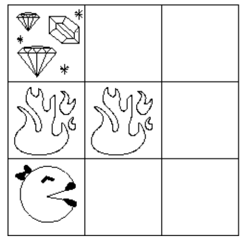
   (a) What are the states and the actions for this MDP?

   (b) What is the transition function and the reward function for this MDP?

   (c) Fill out the value iteration table below. We have filled out the first row for you. (Recall that we always initialize $V_0(s)$ to 0 for all states s.) Then, perform policy extraction and give the optimal policy for this MDP.

V	0	2	3	4	5	Done
V_0	0	0	0	0	0	0
V_1						
V_2						
V_3						
Policy Extraction						

3. Ms.Pacman: While Pacman is busting ghosts, Ms. Pacman goes treasure hunting on GridWorld Island. She has a map showing where the hazards are, and where the treasure is. From any unmarked square, Ms. Pacman can take any of the deterministic actions (N, S, E, W) that doesn't lead off the island. If she lands in a hazard square or a treasure square, her only action is to call for an airlift (X), which takes her to the terminal *Done* state; this results in a reward of -64 if she's escaping a hazard, or +128 if she reached the treasure. There is no living reward.



(a) Let $\gamma = 0.5$. What are the optimal values V^* of each state in the grid above?

(b) Assuming you have V*, how would we compute the Q-values for each state-action pair?

(c) What's the optimal policy? You may use the grid below to fill in the optimal action for each state.

Call this policy π_0.

Ms. Pacman realizes that her map might be out of date, so she uses Q-learning to see what the island is really like. She believes π_0 is close to correct, so she follows an ϵ-random policy, ie., with probability ϵ she picks a legal action uniformly at random (otherwise, she does what π_0 recommends). Call this policy π_ϵ.

π_ϵ is known as a *stochastic* policy, which assigns probabilities to actions rather than recommending a single one. A stochastic policy can be defined with $\pi(s, a)$, the probability of taking action a when the agent is in state s.

(d) Write a modified Bellman update equation for policy evaluation when using a stochastic policy $\pi(s, a)$.

11 Bayes' Nets: Representation, Independence

1. For this problem, any answers that require division can be left written as a fraction.
   PacLabs has just created a new type of mini power pellet that is small enough for Pacman to carry around with him when he's running around mazes. Unfortunately, these mini-pellets don't guarantee that Pacman will win all his fights with ghosts, and they look just like the regular dots Pacman carried around to snack on.
   Pacman (P) just ate a snack , which was either a mini-pellet ($+p$), or a regular dot ($-p$), and is about to get into a fight (W), which he can win ($+w$) or lose ($-w$). Both these variables are unknown, but fortunately, Pacman is a master of probability. He knows that his bag of snacks has 5 mini-pellets and 15 regular dots. He also knows that if he ate a mini-pellet, he has a 70% chance of winning, but if he ate a regular dot, he only has a 20% chance.

   (a) What is $P(+w)$, the marginal probability that Pacman will win?

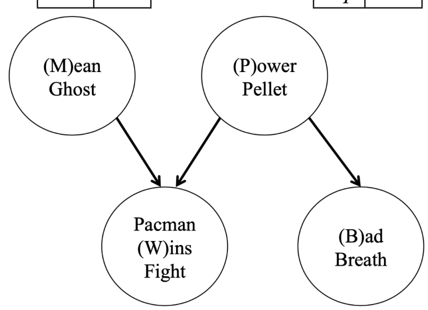
   (b) Pacman won! Hooray! What is the conditional probability $P(+p \mid +w)$ that the food he ate was a mini-pellet, given that he won?

   Pacman can make better probability estimates if he takes more information into account. First, Pacman's breath, B, can be bad ($+b$) or fresh ($-b$). Second, there are two types of ghost (M): mean ($+m$) and nice ($-m$). Pacman has encoded his knowledge about the situation in the following Bayes' Net:

P(M)	
+m	0.50
-m	0.50

P(P)	
+p	0.25
-p	0.75

P(W|M, P)			
+m	+p	+w	0.60
+m	+p	-w	0.40
+m	-p	+w	0.10
+m	-p	-w	0.90
-m	+p	+w	0.80
-m	+p	-w	0.20
-m	-p	+w	0.30
-m	-p	-w	0.70



P(B|P)		
+p	+b	0.80
+p	-b	0.20
-p	+b	0.40
-p	-b	0.60

(c) What is the probability of the event $(-m, +p, +w, -b)$, where Pacman eats a mini-pellet and has fresh breath before winning a fight against a nice ghost?

For the remaining of this question, use the half of the joint probability table that has been computed for you below:

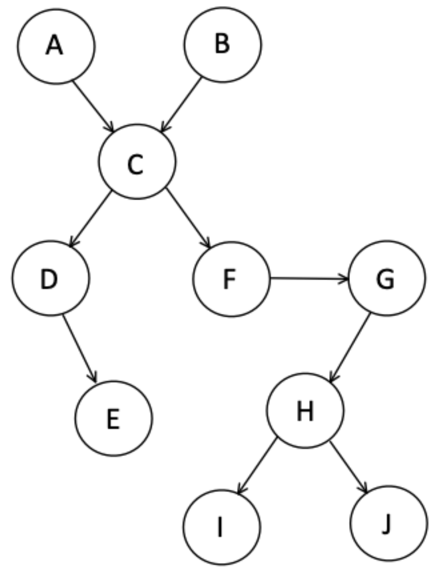
$P(M, P, W, B)$				
$+m$	$+p$	$+w$	$+b$	0.0800
$+m$	$+p$	$+w$	$-b$	0.0150
$+m$	$+p$	$-w$	$+b$	0.0400
$+m$	$+p$	$-w$	$-b$	0.0100
$+m$	$-p$	$+w$	$+b$	0.0150
$+m$	$-p$	$+w$	$-b$	0.0225
$+m$	$-p$	$-w$	$+b$	0.1350
$+m$	$-p$	$-w$	$-b$	0.2025

(d) What is the marginal probability, $P(+m, +b)$ that Pacman encounters a mean ghost and has bad breath?

(e) Pacman observes that he has bad breath and that the ghost he's facing is mean. What is the conditional probability, $P(+w \mid +m, +b)$, that he will win the fight, given his observations?

(f) Pacman's utility is +10 for winning a fight, -5 for losing a fight, and -1 for running away from a fight. Pacman wants to maximize his expected utility. Given that he has bad breath and is facing a mean ghost, should he stay and fight, or run away? Justify your answer.

2. Consider the following Bayes Net:



Determine whether the following statements hold given this Bayes Net

(a) $P(A, B) = P(A)P(B)$

(b) $A \perp\!\!\!\perp E|C$

(c) $D \perp\!\!\!\perp J|I$

(d) $A \perp\!\!\!\perp B|H$

(e) $P(E|C) = P(E|C, G)$

(f) $I \perp\!\!\!\perp J|G$

(g) $P(B|F) = P(B|E, F)$