Short Questions

1. Fill in the blanks

(2 marks)

We learned that some uninformed search algorithms are a variant of the **Best-first Search** algorithm. For example, the Uniform Cost Search algorithm is a form of Best-first Search with f(*n*) = g(*n*).

Determine if the following two uninformed search algorithms may be formulated as a variant of the Best-first Search algorithm, and if so, determine a formulation of f(*n*) such that it does not change their functionality. You are given a helper function **depth(*n*)**, which returns the depth of the node *n* (e.g., f(*n*) = 2 * depth(*n*)). You can assume all step costs are positive and equal, and that the depth at the root is 0.

Breadth-First Search

The Breadth-First Search algorithm may be formulated as a variant of the Best-First Search algorithm (answer only "Y" or "N"): __1__

The formulation of f(*n*) for the Breadth-First Search algorithm (enter "NA" if the answer to your previous question is "N") is: f(*n*) = __2__

Depth-First Search

The Depth-First Search algorithm may be formulated as a variant of the Best-First Search algorithm (answer only "Y" or "N"): __3__

The formulation of f(*n*) for the Depth-First Search algorithm (enter "NA" if the answer to your previous question is "N") is: f(*n*) = __4__

Enter the correct answer below.

1 Character Limit: 1

2 Character Limit: 30

3 Character Limit: 1

4 Character Limit: 30

Response Rationale/Workings

Please provide rationale or workings for your answer.

Font | Size | Source

2. We are given three admissible heuristics: $h_1(s)$, $h_2(s)$ and $h_3(s)$. Determine which, among the following heuristics (i.e., which among $h_4(s)$, $h_5(s)$, $h_6(s)$, $h_7(s)$, $h_8(s)$) are **guaranteed** to be admissible.

Note: select ***all*** options that are valid.

(2 marks)

- [ ] $h_4(s) = h_1(s) + h_2(s) - h_3(s)$
- [ ] $h_5(s) = h_1(s)/4 + h_2(s)/3 + h_3(s)/2$
- [ ] $h_6(s) = \text{MAX}(h_1(s), h_2(s), h_3(s))$
- [ ] $h_7(s) = \text{MIN}(h_1(s) + h_2(s), h_3(s))$
- [ ] $h_8(s) = h_1(s)/3 + h_2(s)/3 + h_3(s)/3$

Response Rationale/Workings

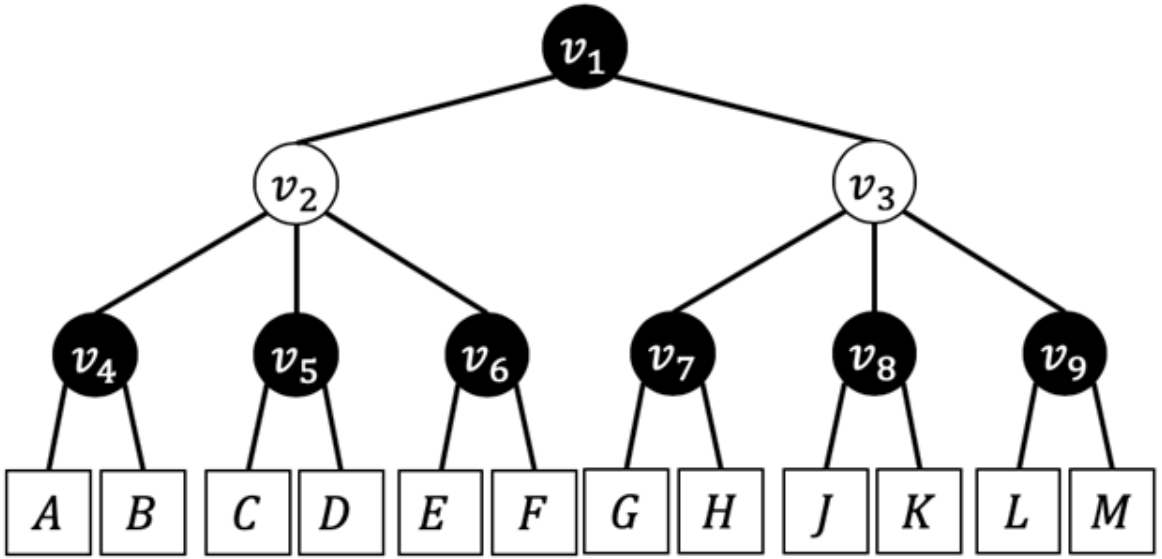
Please provide rationale or workings for your answer.

Font | Size | Source

3. Fill in the blanks

(2 marks)

Consider the following game tree, where black nodes correspond to the MAX player, and white nodes correspond to the MIN player.



Assume that the utility values assigned to the leaf nodes (i.e., nodes A - M) achieve maximum pruning (i.e., given these leaf node values, *the number of leaf nodes that are not evaluated is maximal*). Given this assumption, determine the total number of leaf nodes that are ***not evaluated*** when the Alpha-Beta Pruning algorithm is used.

Note: you must assume that the tree is processed from **right to left**.

Answer: 1

Enter the correct answer below.

1 [ ] Please enter a number for this text box.

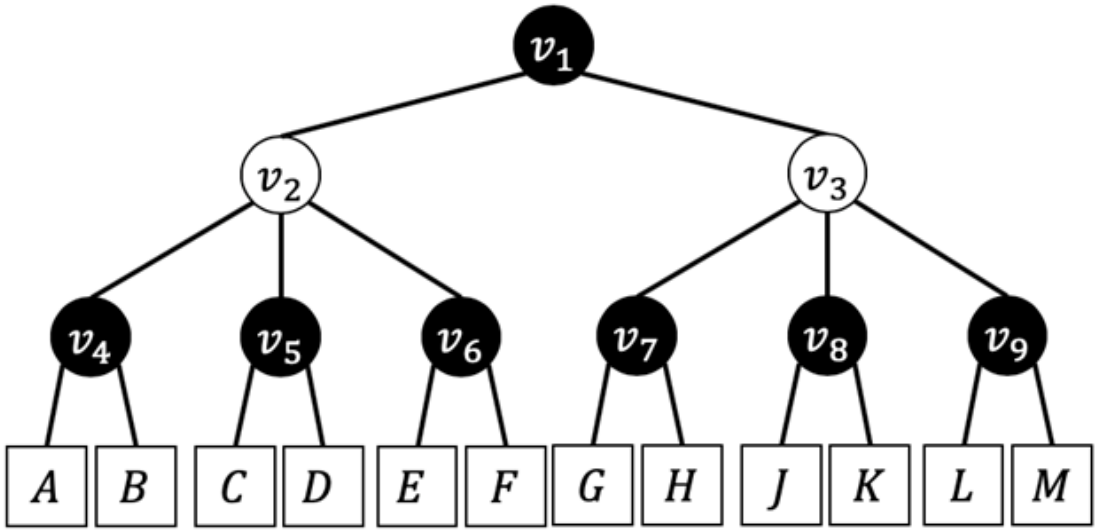
Character Limit: 2

Response Rationale/Workings

Please provide rationale or workings for your answer.

Font | Size | Source

4. Consider the following game tree, where black nodes correspond to the MAX player, and white nodes correspond to the MIN player.



Suppose that the Alpha-Beta Pruning algorithm is applied to the above tree, with traversal going from **right to left**.

You are to assume that any pruning of nodes only occurs due to the application of the **Alpha-Beta Pruning** algorithm.

Select the following statements that are guaranteed to be true.

Note: select ***all*** the options that are valid.

(2 marks)

- [ ] Leaf Node L may or may not be evaluated, depending on the value of leaf node M.
- [ ] We may define a set of values for the leaf nodes such that node K is not evaluated.
- [ ] We may define a set of values for the leaf nodes such that node E is not evaluated.
- [ ] We may define a set of values for the leaf nodes such that node B is not evaluated.
- [ ] We may define a set of values for the leaf nodes such that no pruning occurs anywhere in the game tree.

Response Rationale/Workings

Please provide rationale or workings for your answer.

Font Size Source

5. Select the option that most accurately classifies which among the following sentences (about Constraint Satisfaction Problems) are True. Note that you are to assume the standard specification of the AC-3 algorithm.

(i) The order in which the arcs are processed from the queue in the AC-3 algorithm affects which values are left in the domains of the variables after the algorithm is executed. For example, given that the AC-3 algorithm processes both arcs (A, B) and (B, A), then popping the arc (A, B) before the arc (B, A) will lead to a different resultant domain for A as compared to when arc (B, A) is popped before (A, B).

(ii) By using the Most-Constrained Variable heuristic in conjunction with the Least-Constraining Value heuristic when applying the Backtracking algorithm, any CSP may be solved in linear time.

(1 mark)

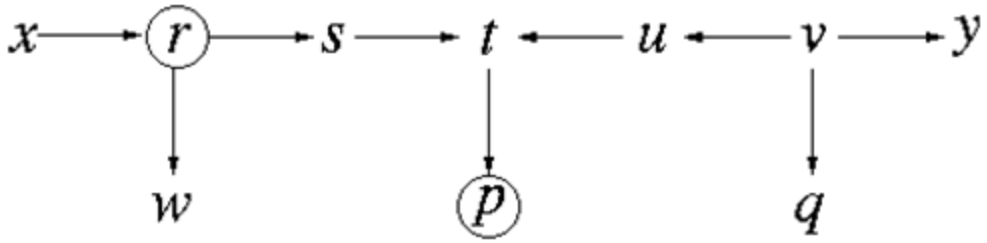
- (i) only.
- (ii) only.
- Both (i) and (ii).
- Neither (i) nor (ii).

Response Rationale/Workings

Please provide rationale or workings for your answer.

Font | Size | Source

6. Consider the Bayesian Network in the picture below (Network *A*), and suppose we are **given** the values of the circled random variables, ***r*** and ***p*.**



Determine if the following statement about *d*-separability is True or False.

In Network *A*, given *r* and *p*, variables *s* and *y* are independent.

(1.5 marks)

True

False

Response Rationale/Workings

Please provide rationale or workings for your answer.

Font | Size | Source

7. Refer to the diagram from Question 6.

Determine if the following statement about *d*-separability is True or False.

In Network *A*, given *r* and *p*, variables *w* and *q* are independent.

(1.5 marks)

True

False

Response Rationale/Workings

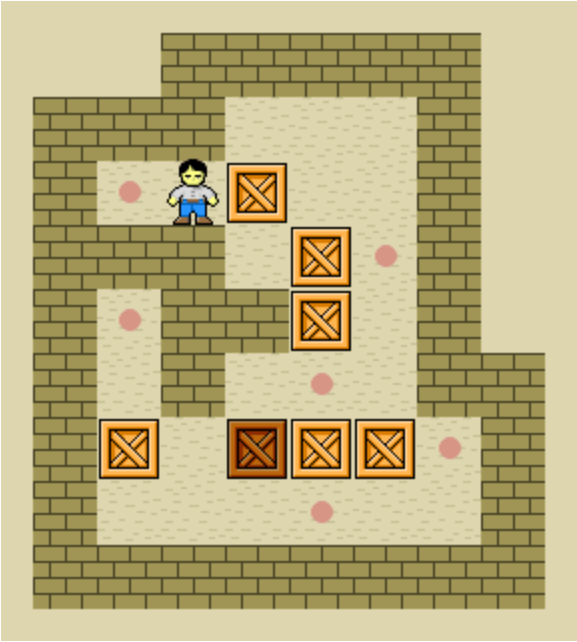
Please provide rationale or workings for your answer.

Font | Size | Source

Save For Later **Next**

Heuristic Design

8. Sokoban is a puzzle game in which the player pushes boxes (or crates) around in a warehouse, trying to get them to specific storage locations. Refer to the GIF below for an example.



A simple example is as follows: assume that there are two boxes, $B_1(x,y)$ and $B_2(x,y)$, in an N × N grid map, M, where some of the grid cells are occupied by immobile obstacles (including walls). Let us further assume that $T_1(x,y)$ and $T_2(x,y)$ represent the cell coordinates corresponding to storage locations, and A(x,y) represents the cell coordinate that contains the agent. ***The objective of the Sokoban game is to move the boxes to the storage locations.***

Each action corresponds to the agent moving up, down, left, or right by one unit. The agent will push a box if it moves into a cell containing a box, and if there is no obstacle in the way - e.g., given A(1,1) and B(1,2), if there is no obstacle or wall at (1,3), then when the agent moves to (1,2), the box at (1,2) is now pushed to (1,3). Do note that an agent may not pull a box.

Based on the given example Sokoban problem, one way to represent a state is thus by defining a tuple (A, M, $B_1(x,y)$, $B_2(x,y)$, $G_1(x,y)$, $G_2(x,y)$). ***Note that we are interested in solving a problem with an arbitrary number of boxes and storage locations***.

Consequently, a goal state is one where every storage location contains a box, and where each box is at a storage location. *Note that there is no requirement for specific boxes to be on top of specific goal positions*.

The following are some additional restrictions/rules to further clarify the game:

1. The agent can neither push any crate beyond the borders of the map, nor move beyond the borders of the map.
2. The crates cannot stack on top of each other - i.e., no two crates can be at the same (x,y) coordinate. This condition also applies to the initial state.
3. The agent only has enough strength to push one box at a time - e.g., given two crates at coordinate (1,2) and (1,3) respectively, and that the agent is located at coordinate (1,1) of the map, the agent will not be able to push the two crates upwards at once (i.e., such that the crates end up at (1,3) and (1,4) respectively) even though there is no obstacle at (1,4).
4. The agent cannot be on the same (x, y) grid as any crate at any point in time.
5. No agent, box or storage location may reside in the same cell as an obstacle at any point in time.

To solve an instance of a Sokoban using informed search, **we may define a heuristic function *h* by relaxing the problem and giving our agent a superpower to move a single box from any one location to another (e.g., from its initial cell coordinate to that of a storage location) in one move (i.e., with a single action).**

Is the heuristic function *h* admissible? Provide a brief rationale to support your claim.

(1 mark)

Enter your answer here

9. With reference to the context given in Question 8, design a new admissible heuristic function, ***h'***, that dominates the heuristic function ***h*** (note that ***h*** was defined in Question 8).

More specifically, you must:

(a) Define a new admissible heuristic, *h'*, which dominates *h*;
(b) Show that *h'* is admissibility; and
(c) Show that *h'* dominates *h.*

Note that marks will only be awarded for (b) and (c); there are no marks directly assigned for (a).

(6 marks)

Enter your answer here

Previous Save For Later **Next**

Constraint Satisfaction Problems

10. Recall the election model introduced in the midterm quiz:

We have a set of n voters in V = {v_1, …, v_n} and a set of m candidates in C = {c_1, …, c_m}, whereby each voter in V possesses a full strict preference order over all candidates in C. More formally, each voter v_i in V has an internal preference ranking of candidates $x_{i1} > x_{i2} > \ldots > x_{im}$, where x_{ij} is the candidate that v_i (i.e., voter i) ranked j-th in its preference ranking.

Based on these individual preferences, the election committee will select a winner *w* from C using *some* voting process/algorithm.

We would like to model *certain aspects of the above setting* as a constraint satisfaction problem. The following are **constraints to be modelled**.

1. **UniquePreference** - this constraint tests, given any voter, if the candidates in its preference lists are unique (i.e., no repetition).
2. **GoodWinner** - this constraint tests, given a winner, *w* (chosen from the set of candidates by the algorithm), the following properties:
    - The winner *w* must be ranked first in at least one voter's preference list.
    - The winner *w* cannot be ranked last in any voter's preference list.

To assist you with the definition of the above two constraints, you are also given the following functions:

- **pref(v, k)**, which returns the k-th element of an ordered tuple of m-elements, which in this case corresponds to the candidate ranked k-th in voter v's preference ranking.
- **rank(v, c)**, which returns the rank of candidate c in voter v's preference list.

Model each of the above two constraints using only the given variables (in the question), the helper function(s), *standard* logical operators (for all, there exists, and, or, etc.) or *standard* mathematical operations (+, -, x, /, >, >=, <, <=, !=, =, sum, product, etc.).

(7 marks)

Enter your answer here

Response Rationale/Workings

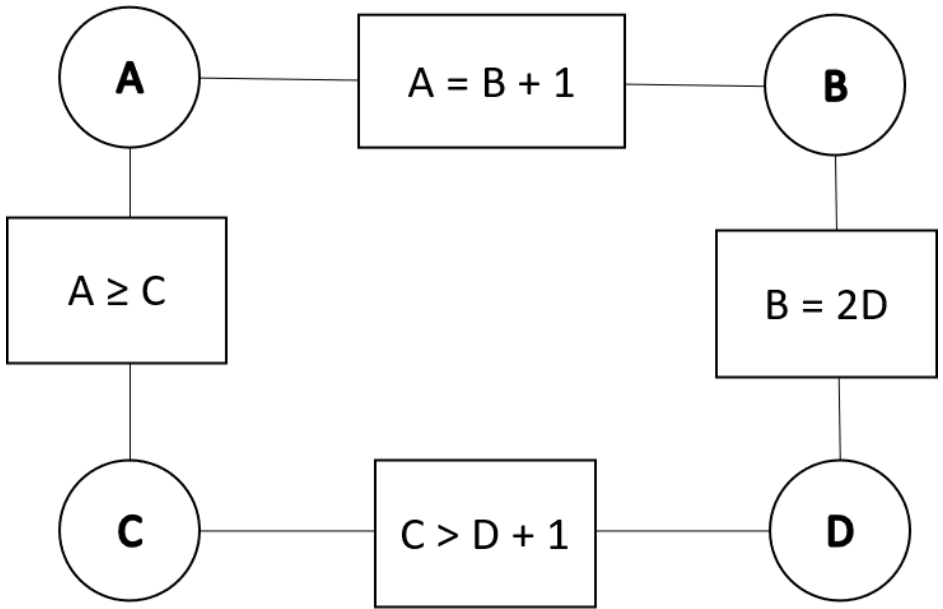
Please provide rationale or workings for your answer.

Font | Size | Source

11. Fill in the blanks

(4 marks)

Suppose that we are in the midst of executing the AC-3 algorithm on a CSP whose constraint graph is depicted below.



Complete the execution of the AC-3 algorithm. You are to assume that the current queue and the current domains of each variable given below.

Current Queue: (B, D), (D, B), (D, C), (C, D)

Current Domains of each variable:

D_A: {2, 3, 4, 5}

D_B: {1, 2, 3, 4}

D_C: {1, 2, 3, 4, 5}

D_d: {1, 2, 3, 4, 5}

(a) Provide your trace of the algorithm in the rationale box. Note that if the arc is already in the queue, do NOT add it to the queue again.

The format of your AC-3 algorithm trace should be something like this (just an example):

Arc popped; Any domains pruned; Queue at the end the iteration.

1. Pop (P, Q); No Domains Pruned; Queue: (Y, X), (Z, W)

2. Pop (Y, X); D_Y (i.e., Domain of Y) = (117,118); Queue: (Z, W)

…

(b) State the values in the domain of each variable after running AC-3 on the constraint graph in the blanks below.

The format of the domain values is as such: (x,y,z) (NO WHITESPACES IN BETWEEN) For example, if the remaining values left in the domain of A are 10 and 11, your answer should be this: (10,11).

Domain of A after running AC-3: 1

Domain of B after running AC-3: 2

Domain of C after running AC-3: 3

Domain of D after running AC-3: 4

As mentioned earlier, remember to enter the trace of the AC-3 algorithm into the rationale box.

Enter the correct answer below.

1

2

3

4

Response Rationale/Workings

Please provide rationale or workings for your answer.

Font | Size | Source

12. Fill in the blanks

(1 mark)

With reference to the context given in Question 11, provide a valid assignment of values to A, B, C and D such that all the earlier specified constraints are satisfied, and also the additional constraint A + B + D = 2C + 1

A: 1

B: 2

C: 3

D: 4

Enter the correct answer below.

1 Please enter a number for this text box.

2 Please enter a number for this text box.

3 Please enter a number for this text box.

4 Please enter a number for this text box.

Response Rationale/Workings

Please provide rationale or workings for your answer.

Font Size Source

Logical Agents

13. Ancient Lore in the World of Adventure tells us that:

1. **Every dragon sleeps in some lair.**
2. **Every wyvern is a dragon, and every wyvern is poisonous.**
3. **Every lair in which a poisonous dragon sleeps is toxic.**
4. **Anything that sleeps in anything that is toxic has slime minions.**

Note: the above are to be taken as facts in the World of Adventure.

A wizard now claims that **every wyvern has slime minions**.

Using resolution, prove the wizard's claim. You are to trace out your resolution steps in the answer box below.

Note: there will be no Error Carried Forward (ECF) considered for this question.

Hint: you should **NOT** use first-order logic (FOL)

(5 marks)

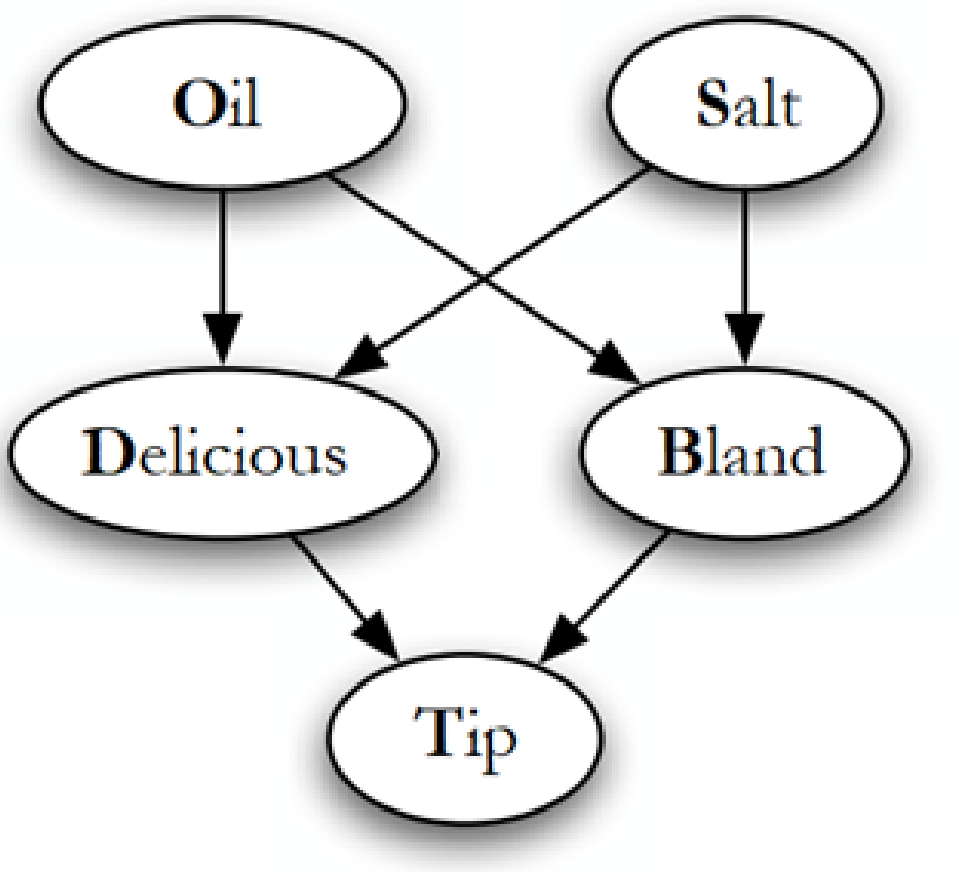
Enter your answer here

Bayesian Networks

14. An ambitious restaurant waiter wants to maximise the number of tips he earns. He notices that whether patrons tip or not depends on the quality of the chef's cooking. When the chef uses more *Oil* and more *Salt*, more people comment that the food is *Delicious* and fewer people say that the food is *Bland*. When people comment that the food is delicious, the waiter is much more likely to receive a *Tip*.

If the waiter models this by writing down the joint probability distribution table, then the minimum number of table entries required to fully specify the scenario is $2^5 - 1 = 31$.

Suppose that the employee does not want to write down a table that large. Instead, he uses his domain knowledge to represent the scenario as the following Bayesian Network.



Recall that a Bayesian Network represents the full joint distribution in a compact form by exploiting knowledge of which variables are conditionally independent.

Write down the formula that this Bayesian Network expresses for the joint probability distribution, Pr[S, B, D, T]. Note that O is intentionally omitted from the joint distribution. Ensure that your answer contains only terms that are obtainable from the Conditional Probability Tables (CPT) given by the Bayesian network.

(2 marks)

Enter your answer here

Word Limit:

15. Assuming the context from Question 16, determine if the following statement is **True** or **False.**

Given the values of **O**il and **S**alt, **D**elicious is conditionally independent of every other variable.

(1 mark)

- True
- False

16. Assuming the context from Question 16, determine if the following statement is **True** or **False.**

Given the values of **D**elicious and **B**land, **T**ip is conditionally independent of every other variable.

(1 mark)

- True
- False

17. Fill in the blanks

(1 mark)

Suppose that for the Bayesian Network given in Question 16, we are given the following probabilities:

Pr[O] = 0.4
Pr[S] = 0.2

O	S	Pr[D | O, S]
F	F	0.2
F	T	0.4
T	F	0.3
T	T	0.6

O	S	Pr[B | O, S]
F	F	0.7
F	T	0.4
T	F	0.4
T	T	0.1

D	B	Pr[T | D, B]
F	F	0.2
F	T	0.1
T	F	0.6
T	T	0.3

(Note: **D**elicious and **B**land are not opposites of each other. Some people at the table might comment that the food is delicious, whereas others might comment that it is bland. The **T**ip probability is expressed in terms of whether at least one D/B comment was heard.)

Compute the following probability (give the answer as a decimal value, e.g., 0.3; express your answer to 3 significant figures).

Probability that a **T**ip was offered, given that someone said the food is **B**land.

Probability: 1

Enter the correct answer below.

1 Please enter a number for this text box.

18. Fill in the blanks

(2 marks)

Based on the Bayesian Network given in Question 16, and the CPTs given in Question 19. Compute the following probability (give the answer as a decimal value, e.g., 0.3; express your answer to 3 significant figures).

Probability that someone said the food was **D**elicious.

Probability: 1

Enter the correct answer below.

1 Please enter a number for this text box.

19. Fill in the blanks

(2 marks)

Based on the Bayesian Network given in Question 16, and the CPTs given in Question 19. Compute the following probability (give the answer as a decimal value, e.g., 0.3; express your answer to 3 significant figures).

Probability that the chef added **O**il, given that someone said the food was **D**elicious.

Probability: 1

Enter the correct answer below.

1 Please enter a number for this text box.