National University of Singapore

CS2109S—Introduction to AI and Machine Learning

MIDTERM ASSESSMENT

Semester 2, 2022/2023

Time allowed: 2 hours

INSTRUCTIONS TO STUDENTS

1. Write down your **Student Number** on the answer sheet and shade completely the corresponding bubbles in the grid for each digit or letter. **DO NOT WRITE YOUR NAME!**
2. The assessment paper contains **FIVE (5) questions** and comprises **THIRTEEN (13) pages** including this cover page.
3. Weightage of questions is given in square brackets. The maximum attainable score is 100.
4. This is an **<u>OPEN-SHEET</u>** assessment, which means that you are permitted to use a double-sided A4-sized cheatsheet.
5. You are allowed to bring a calculator, but it cannot have any form of external communication capability, i.e. not Wifi- or 4G-enabled. Mobile phones and tablets are not allowed.
6. All questions must be answered in the space provided on the answer sheet; no extra sheets will be accepted as answers.
7. You are allowed to write with pencils, as long as it is legible.
8. **Marks may be deducted** for unrecognisable handwriting and/or for not shading the student number properly.
9. You must submit only the **ANSWER SHEET** and no other documents. The question set may be used as scratch paper.
10. An excerpt of the question may be provided in the answer sheet to aid you in answering in the correct box, where applicable. It is not the exact question. You should still refer to the original question in this question booklet.

This page is intentionally left blank.

It may be used as scratch paper.

Question 1: Frogger [30 marks]

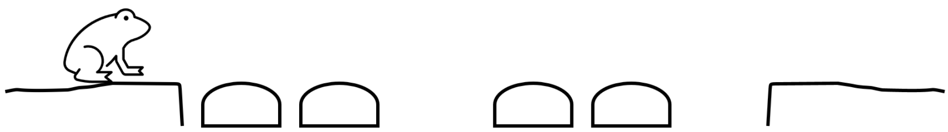


Figure 1: Frog Puzzle

Consider the following puzzle, where we have a frog that is trying to move from one side of a river to the other side without falling into the water. Between the 2 banks of a river are n "steps." Some of these steps are rocks (that the frog can land on), while the remaining steps are gaps between the rocks and the frog will end up falling into the river if it lands in those gaps.

You can assume that the you have as input an array of n characters, where `"0"` represents a rock and `"_"` represents gap. Hence:

(a) `[0, 0, 0, 0, 0]`: would be a 5-step-wide river where there is a rock at every step.

(b) `[0, 0, _, _, 0, 0, 0]`: would be a 7-step-wide river with a 2-step gap in the middle.

In each move, the frog can either walk one step, or make a jump of exactly k steps ahead, where $k \geq 2$. The frog cannot remain in the same position.

Warm Up

Your goal is to come up with a mimimal (shortest) sequence of moves that will allow the frog to cross the river without falling into the water.

Suppose $k = 2$, then the solution to (a) would be `[2,2,2]` and there is no solution to (b), since the frog will not be able to jump across the 2-step gap.

Suppose $k = 3$, then the solution to (a), would be `[3,3]` and the solution to (b) would be `[1,1,3,3]`.

A. Propose a state (minimal) representation for this problem if we want to formulate the solution as a search problem and define the corresponding actions. You can assume that the input array is a constant and hence does not need to be included in your representation.

[2 marks]

B. What are the initial and goal states for the problem under your proposed representation in Part (A)? [2 marks]

C. Which of the following statement(s) is/are true given your definition of the search problem in Parts (A) to (B)? Shade <u>all</u> that is/are true. [3 marks]

- The search tree is finite.

- There are possibly many repeated states.
- There are possibly multiple goal states.
- We can always find an answer (valid sequence of moves) if we employ the right search algorithm.
- None of the above.

D. Suppose we decide to apply TREE-SEARCH (See Appendix) using one of the following algorithms:

1. Depth-first search (DFS)
2. Breath-first search (BFS)
3. Depth-limited search (DLS)
4. Iterative-deepening search (IDS)

Which of the above search algorithms should we choose? Note that you might have to make some minor enhancements/modifications, if needed. Explain. [2 marks]

Variant #1: Variable Jumps & Exact Moves

Instead of coming up with a mimimal (shortest) sequence of moves, you are now asked to find a sequence of exactly m moves that will allow the frog to cross the river without falling into the water. Also, instead of being limited to jumps of exactly k steps, the frog can make jumps up to k steps, $k \geq 1$. Note that in this problem, the frog might need to jump backwards(!).

E. Propose a state representation for this problem if we want to formulate it as a search problem and define the corresponding actions. You can assume that the input array is a constant and hence does not need to be included in your representation. [2 marks]

F. What are the initial and goal states for the problem under your proposed representation in Part (E)? [2 marks]

G. What is the invariant for your state representation in Part (E) above? In other words, what are the condition(s) that the state representation must satisfy, in order to be valid? [2 marks]

H. Which of the following statement(s) is/are true given your definition of the search problem in Parts (E) to (G)? Shade all that is/are true. [3 marks]

- The search tree is finite.
- There are possibly many repeated states.
- There are possibly multiple goal states.

- We can always find an answer (valid sequence of moves) if we employ the right search algorithm.
- None of the above.

I. Suppose we decide to apply TREE-SEARCH (See Appendix) using one of the following algorithms:

1. Depth-first search (DFS)
2. Breath-first search (BFS)
3. Depth-limited search (DLS)
4. Iterative-deepening search (IDS)

Which of the above search algorithms should we choose? Note that you might have to make some minor enhancements/modifications, if needed. Explain. [2 marks]

J. Should we have used GRAPH-SEARCH (See Appendix) in Part (I) instead of TREE-SEARCH? Explain. [2 marks]

Variant #2: Takes Energy to Jump & Energy is Limited

Finally, you are asked not only to find a sequence of exactly m moves when the frog can make jumps up to k steps, you are given a constraint that the frog only has E units of energy at the start. To make a move of k steps, the frog will consume k^2 units of energy. You are to find a sequence that minimizes the total amount of energy required for the frog to cross the river in exactly m moves.

K. Explain how the state representation and actions in Part(E) need to be modified to solve this modified problem. [2 marks]

L. Which of the following statement(s) is/are true given your definition of the modified search problem in Part (K)? Shade all that is/are true. [3 marks]

- The search tree is finite.
- There are possibly many repeated states.
- There are possibly multiple goal states.
- We can always find an answer (valid sequence of moves) if we employ the right search algorithm.
- None of the above.

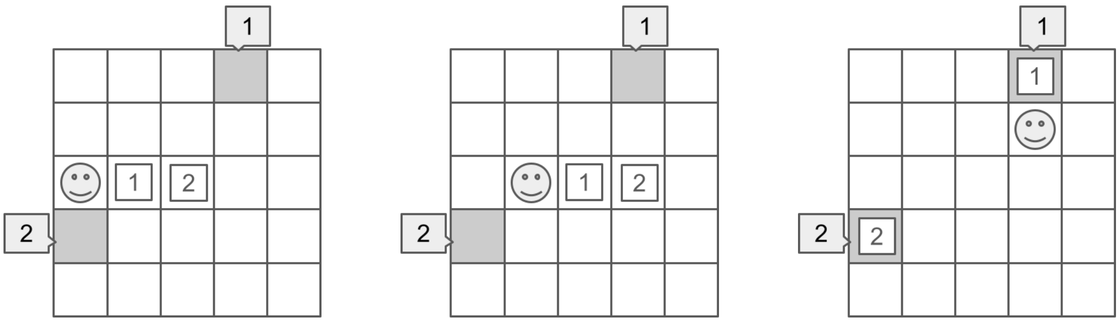
M. Which search algorithm would you use to solve the modified problem in Part (K)? Can we always determine if a solution can be found? Explain. [3 marks]

Question 2: Sokoban [28 marks]

Sokoban is a classic puzzle game where you control a character who pushes boxes onto designated goal positions in a 2-dimensional grid. In this question, we consider a variant of Sokoban which has a grid size of $W \times H$, N boxes, and N goal positions where $1 \leq N \ll W \times H$, $W > 0$, and $H > 0$. Each box and goal position is numbered from 1 to N. At the beginning of the game, the boxes are scattered throughout the grid and the character must move them to their corresponding goal positions. Here, we assume that the initial state of the Sokoban is solvable, but is not solved yet.

At each time interval, the character can move in four cardinal directions: up, down, left, and right, with each movement incurring a cost of 1. When the character moves into a position where there is a box, the box is pushed in the same direction. For example, if the character is at position $(x,y) = (0,2)$ and there is a box at position $(1,2)$, moving the character to the right will result in the character's position becoming $(1,2)$ and the box being moved to position $(2,2)$. If there are other boxes adjacent to the box being pushed, they will also move along with it. However, neither the character nor the boxes can move beyond the boundaries of the grid.

The objective of the game is to move each numbered box to its corresponding goal position, which has the same number. For example, if there are two boxes numbered 1 and 2, the character must move box 1 to goal position 1 and box 2 to goal position 2.



(a) Initial state (b) Move right from the initial state (c) Goal state

Figure 2: 5×5 **Sokoban with two boxes and goals**

Your cousin Ben Bitdiddle immediately suggests using similar heuristics that he has seen in the CS2109S lectures.

For each heuristic, **shade** the appropriate circle to indicate if the heuristic is admissible/consistent or not, then justify your answer in the boxes provided.

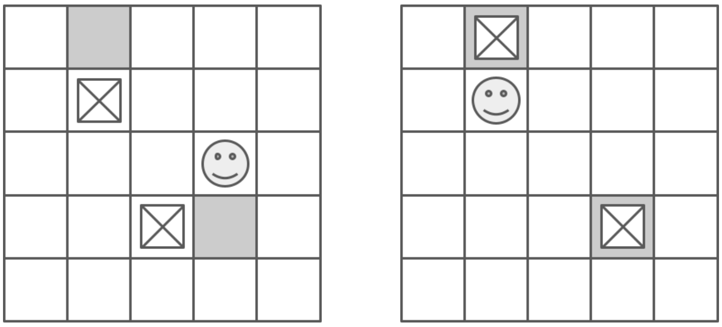
Hint: Manhattan Distance (MD) is $|x_1 - x_2| + |y_1 - y_2|$

A. **[Warm Up]** h_1 : number of misplaced boxes [2 marks]

B. h_2 : number of misplaced boxes divided by the total number of boxes [2 marks]

Let's consider a modified version of the Sokoban game where there are N boxes and goals, and they are not numbered. In this variant, **the character can push any box to any goal**.

For instance, suppose there are two boxes and two goals. The objective of the game is still to move each box to a goal position, but now the player has more freedom to choose which box to push to which goal.



(a) Unfinished state (b) Goal state

Figure 3: Modified 5×5 **Sokoban with non-numbered boxes and goals**

Ben Bitdiddle has come up with the following possible heuristics to be used for this new variant of Sokoban.

Let p be the position of the character, B be the set of positions of boxes and G be the set of goal positions of the boxes.

C. **[Warm Up]** h_A : number of misplaced boxes [2 marks]

D. Minimum of all distances of each box to their farthest goal position. [6 marks]

$$h_B : \min_{b \in B}\{\max_{g \in G}\{\text{MD}(b,g)\}\}$$

E. Maximum of all distances of each box to their closest goal position. [6 marks]

$$h_C : \max_{b \in B}\{\min_{g \in G}\{\text{MD}(b,g)\}\}$$

F. The gap between the character to its closest box, plus the distance from the box to its closest goal. Tie-breaker: choose the box with the lowest distance to the goal. [6 marks]

$$h_D : \big(MD(p,b) - 1\big) + \min_{g \in G}\{MD(b,g)\}, \text{ where } b = \arg\min_{b' \in B}\{MD(p,b')\}$$

G. **[Dominance]** State and explain the dominant relationships between the proposed heuristics in parts C-F.

[4 marks]

Question 3: Predicting Housing Prices [24 marks]

Your uncle Ben Bitdiddle, a quirky entrepreneur, dreams of launching a startup that predicts housing prices. For this, he has sketched out a decision tree as the prediction model (Figure 4).

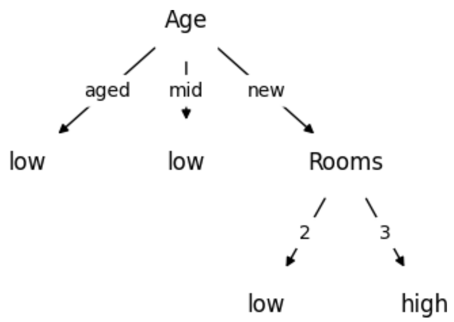


Figure 4: Ben Bitdiddle's Decision Tree

Please note that showing your working is not required for this question. However, if you wish to do so, you may include it just in case. All multiple-choice questions will be automatically graded and we will not review your working, unless you submit an appeal due to an error (such as shading incorrectly by mistake).

A. **[Warm Up]** A potential client is interested in the price of a 2-room house. Based on the decision tree in Figure 4, what is the predicted price (High or Low)? [2 marks]

You're skeptical of Uncle Ben's decision tree for predicting housing prices. To help him, you gathered a set of data which is shown in Table 1 below. You're determined to prove that the model needs improvement, and you're ready to roll up your sleeves and dive into the numbers.

	Age	**Rooms**	**Location**	**Condition**	**Price**
0	aged	2	good	excellent	high
1	aged	2	good	poor	low
2	mid	2	good	excellent	high
3	mid	3	bad	poor	low
4	mid	3	good	poor	high
5	new	2	good	poor	high
6	new	2	good	poor	high
7	new	3	bad	poor	low
8	new	3	bad	poor	low
9	new	3	good	excellent	high

Table 1: Housing Prices Data

The information content for a given probability distribution p_i, for $i = 1, \ldots, n$ is given by:

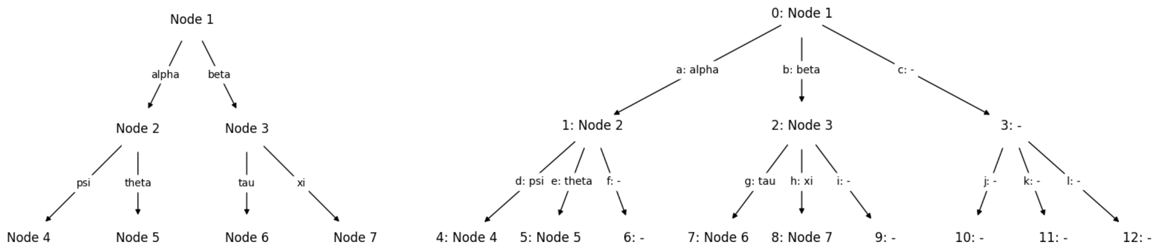
$$\text{Entropy} = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

B. What is the entropy of the outcomes (High/Low) in Table 1, rounded to 2 decimal places? *Hint: remember the* log *is in base 2!* [2 marks]

C. To quickly demonstrate to your uncle that his decision tree is flawed, you decided to create a one-level decision tree (with only one split) using information gain for comparison. [6 marks]

Complete the decision tree in the given template by answering the multiple choice questions. Make sure to use the appropriate calculations to choose the attributes to split.

<u>IMPORTANT:</u> You need to complete the decision tree in the provided template in a specific way. In particular, please refer to Figure 5 below for an example of how we expect you to represent your answer. After you have derived the required decision tree, please arrange the edges in **ascending** alphanumerical order. If there is no node/edge in a particular location, use a "-" symbol. When using the "-" symbol, place it **on the right side** of existing nodes/edges. This means that there should be no "-" symbol on the left side of any nodes/edges within the same branch.



(a) Decision Tree (b) Completed Decision Tree using our Template

Figure 5: An example on how to complete the decision tree in the provided template.

D. Your uncle is not convinced by your decision tree because it is too simple. To convince him, you have decided to construct a full decision tree.

Suppose that you pick "Location" attribute as the root of your decision tree. Using the data in Table 1 and information gain to split the data, create the remaining decision tree for "Location" = "good". In case of a tie, the priority order for constructing the tree is Location > Age > Condition > Rooms. [6 marks]

E. To further prove that your uncle's decision tree is inadequate, you plan to evaluate its performance on real-world data.

Using the data in Table 1, Figure out the true positives, false positives, true negatives, and false negatives of your uncle's decision tree (Figure 4). Then, calculate the precision, recall, and F1 score. Use "High" as a positive label and "Low" as a negative label. [5 marks]

F. **[Min-Sample Pruning]** Suppose you want to prune the decision tree you constructed in Part (D) to have at least 3 training data points per leaf. What is the pruned decision tree?

[3 marks]

Question 4: Solving Games [15 marks]

A. **[Warm Up]** Consider a game where two players take turns moving a peg towards the opposite end of a 1-dimensional grid of length 7. At the beginning, the pegs are positioned on the opposite ends of the board. At each turn, the player on the **left** side of the grid (first player) can move their peg 1 or 2 steps **to the right**, while the player on the **right** side of the grid (second player) can move their peg 1 or 2 steps **to the left**. The pegs cannot jump over each other and cannot be in the same position. The game ends when a player is unable to move their peg. At that point, the other player is declared the winner (i.e., the player who cannot move **loses**).

For example, consider a state of the game where the left peg is at position 1 and the right peg is at position 7. The first player can move their peg 1 or 2 steps to the right, so they might choose to move to position 2 or position 3. The second player can then move their peg 1 or 2 steps to the left, so they might choose to move to position 5 or position 6. The game continues in this way until one player is unable to move their peg, at which point the other player wins.

Consider the **full** game tree that solves this problem and answer the following questions:

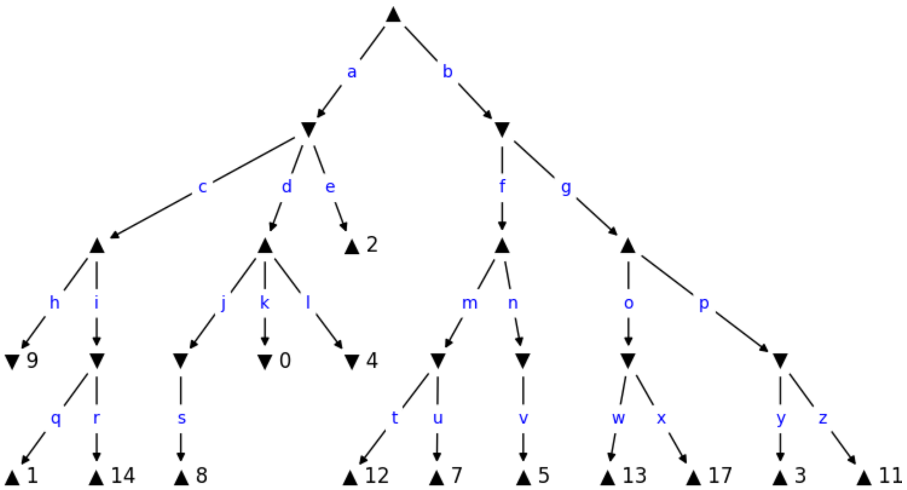
- What is the minimum number of variables required to maintain the game state?
- How many nodes (not including the leaves) are there in the game tree?
- What is the number of leaves where the first player wins?
- What is the number of leaves where the first player loses?
- What is the total number of edges in the game tree?
- Which player do we expect to win?

[6 marks]

Note: Please ensure you choose the most efficient representation, i.e., a state representation uses the minimal number of variables, for your game tree. Also, ensure that all leaves are terminal states.

Please note that showing your working is not required for this question. However, if you wish to do so, you may include it just in case. All multiple-choice questions will be automatically graded and we will not review your working, unless you submit an appeal due to an error (such as shading incorrectly by mistake).

B. **[Alpha-beta left-to-right]** In lecture, we discussed *Alpha-beta pruning*. Consider the following minimax tree:



Suppose we traverse this tree with DFS from left to right. Shade <u>all</u> the link(s) that would be pruned by *alpha-beta*. Select only the links that are <u>directly</u> pruned by alpha-beta and not those that are indirectly pruned because they are in a subtree of a pruned link. Indicate the final value of the root node. [5 marks]

C. **[Alpha-beta right-to-left]** Suppose we traverse the minimax tree in Part (B) with DFS from right to left instead. Shade <u>all</u> the link(s) that would be pruned by *alpha-beta*. [4 marks]

Question 5: Reflections (Free marks!) [3 marks]

What are the 3 most important lessons that you think you learnt in CS2109S thus far? Explain. [3 marks]

Appendix

The following is one of the algorithms that was introduced in class that is reproduced here for your reference.

function TREE-SEARCH(*problem,frontier*) **returns** a solution, or failure

frontier ← INSERT(MAKE-NODE(INITIAL-STATE[*problem*]),*frontier*)

loop do

if *frontier* is empty **then return** failure

node ← REMOVE-FRONT(*frontier*)

if GOAL-TEST[*problem*] applied to STATE(*node*) succeeds **return** *node*

frontier ← INSERTALL(EXPAND(*node*, *problem*),*frontier*)

function GRAPH-SEARCH(*problem,frontier*) **returns** a solution, or failure

closed ← an empty set

frontier ← INSERT(MAKE-NODE(INITIAL-STATE[*problem*]),*frontier*)

loop do

if *frontier* is empty **then return** failure

node ← REMOVE-FRONT(*frontier*)

if GOAL-TEST(*problem*, STATE[*node*]) **then return** *node*

if STATE[*node*] is not in *closed* **then**

add STATE[*node*] to *closed*

frontier ← INSERTALL(EXPAND(*node*, *problem*),*frontier*)

end

function DECISION-TREE-LEARNING(*examples, attributes, default*) **returns** a decision tree

inputs: *examples*, set of examples

attributes, set of attributes

default, default value for the goal predicate

if *examples* is empty **then return** *default*

else if all *examples* have the same classification **then return** the classification

else if *attributes* is empty **then return** MAJORITY-VALUE(*examples*)

else

$best \leftarrow$ CHOOSE-ATTRIBUTE($attributes, examples$)

$tree \leftarrow$ a new decision tree with root test *best*

for each value v_i of *best* **do**

$examples_i \leftarrow \{$elements of *examples* with $best = v_i\}$

subtree $\leftarrow$ DECISION-TREE-LEARNING($examples_i, attributes - best$,

MAJORITY-VALUE(*examples*))

add a branch to *tree* with label v_i and subtree *subtree*

end

return *tree*

— END OF PAPER —

CS2109S—Introduction to AI and Machine Learning

Midterm Assessment Answer Sheet

Semester 2, 2022/2023

Time allowed: 2 hours

Instructions (please read carefully):

1. Write down your **Student Number** on the right and using ink or pencil, **shade completely** the corresponding bubbles in the grid for each digit or letter. **DO NOT WRITE YOUR NAME!**
2. This answer booklet comprises **NINETEEN (19) pages**, including this cover page.
3. This is a **OPEN-SHEET** assessment. You are allow one A4-sized double-sided cheatsheet.
4. Weightage of questions is given in square brackets. The maximum attainable score is 100.
5. You are allowed to bring a calculator, but it cannot have any form of external communication capability, i.e. not Wifi- or 4G-enabled. Mobile phones and tablets are not allowed.
6. All questions must be answered in the space provided on the answer sheet; no extra sheets will be accepted as answers.
7. You are allowed to write with pencils, as long as it is legible.
8. **Marks may be deducted** for unrecognisable handwriting and/or for not shading the student number properly.
9. You must submit only the **ANSWER SHEET** and no other documents. The question set may be used as scratch paper.
10. An excerpt of the question may be provided to aid you in answering in the correct box, where applicable. It is not the exact question. You should still refer to the original question in the question booklet.

STUDENT NUMBER

A

U
A
HT
NT

For Examiner's Use Only

Question	Marks
Q1	/ 30
Q2	/ 28
Q3	/ 24
Q4	/ 15
Q5	/ 3
Total	/100

Question 1A State Representation [2 marks]

Question 1B Initial and goal states [2 marks]

Question 1C Which of the following statement(s) is/are true? [3 marks]

- ◯ The search tree is finite.
- ◯ There are possibly many repeated states.
- ◯ There are possibly multiple goal states.
- ◯ We can always find an answer (valid sequence of moves) if we employ the right search algorithm.
- ◯ None of the above.

Question 1D Best search algorithm. Explain. [2 marks]

Question 1E State Representation for Variant #1 [2 marks]

Question 1F Initial and goal states for Variant #1 [2 marks]

Question 1G Invariant for Variant #1 [2 marks]

Question 1H Which of the following statement(s) is/are true for Variant #1? [3 marks]

- ◯ The search tree is finite.
- ◯ There are possibly many repeated states.
- ◯ There are possibly multiple goal states.
- ◯ We can always find an answer (valid sequence of moves) if we employ the right search algorithm.
- ◯ None of the above.

Question 1I Best search algorithm for Variant #1. Explain. [2 marks]

Question 1J To Graph-Search or not? That is the question. [2 marks]

Question 1K State Representation for Variant #2 [2 marks]

Question 1L Which of the following statement(s) is/are true for Variant #2? [3 marks]

- ◯ The search tree is finite.
- ◯ There are possibly many repeated states.
- ◯ There are possibly multiple goal states.
- ◯ We can always find an answer (valid sequence of moves) if we employ the right search algorithm.
- ◯ None of the above.

Question 1M Best search algorithm for Variant #2. Explain. [3 marks]

Question 2A Is h_1 admissible? Explain. [2 marks]

◯ Admissible ◯ Not admissible

Question 2B Is h_2 admissible? Explain. [2 marks]

◯ Admissible ◯ Not admissible

Question 2C Is h_A admissible? Explain. [2 marks]

◯ Admissible ◯ Not admissible

Question 2D h_B admissible and consistent? Explain. [6 marks]

◯ Admissible ◯ Not admissible

◯ Consistent ◯ Not consistent

Question 2E Is h_C admissible and consistent? Explain. [6 marks]

◯ Admissible ◯ Not admissible

◯ Consistent ◯ Not consistent

Question 2F h_D admissible and consistent? Explain. [6 marks]

◯ Admissible ◯ Not admissible

◯ Consistent ◯ Not consistent

Question 2G Dominant relationships between the heuristics. [4 marks]

Which heuristic(s) does h_A dominate? Shade <u>all</u> that apply/ies.

○ h_B ○ h_C ○ h_D ○ None

Which heuristic(s) does h_B dominates Shade <u>all</u> that apply/ies.

○ h_A ○ h_C ○ h_D ○ None

Which heuristic(s) does h_C dominate? Shade <u>all</u> that apply/ies.

○ h_A ○ h_B ○ h_D ○ None

Which heuristic(s) does h_D dominate? Shade <u>all</u> that apply/ies.

○ h_A ○ h_B ○ h_C ○ None

Question 3A The price of a 2-room house? [2 marks]

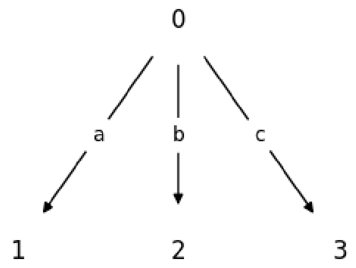
○ High ○ Low ○ Undefined

Question 3B What is the entropy? [2 marks]

○ $0.0 \leq \text{Entropy} < 0.2$ ○ $0.2 \leq \text{Entropy} < 0.4$
○ $0.4 \leq \text{Entropy} < 0.6$ ○ $0.6 \leq \text{Entropy} < 0.8$
○ $0.8 \leq \text{Entropy} \leq 1.0$

Show your working below (if you wish):

Question 3C One-level Tree. [6 marks]



What is node 0?

○ Age ○ Condition ○ Location ○ Rooms
○ High ○ Low ○ Low/High ○ -

What is node 1?

○ Age ○ Condition ○ Location ○ Rooms
○ High ○ Low ○ Low/High ○ -

What is node 2?

○ Age ○ Condition ○ Location ○ Rooms
○ High ○ Low ○ Low/High ○ -

What is node 3?

○ Age ○ Condition ○ Location ○ Rooms
○ High ○ Low ○ Low/High ○ -

What is edge a?

○ 2 ○ 3 ○ aged ○ bad ○ excellent
○ good ○ mid ○ new ○ poor ○ -

What is edge b?

○ 2 ○ 3 ○ aged ○ bad ○ excellent
○ good ○ mid ○ new ○ poor ○ -

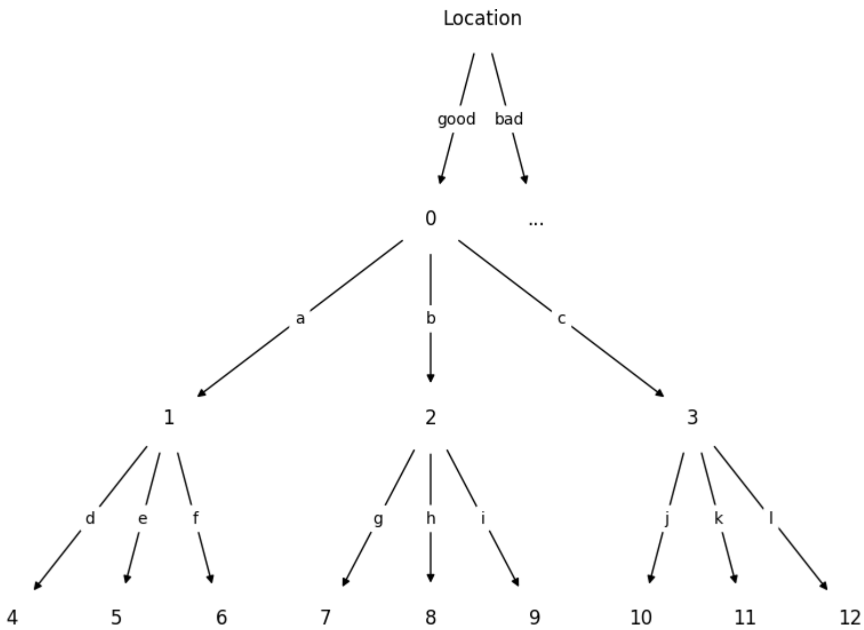
What is edge c?

○ 2 ○ 3 ○ aged ○ bad ○ excellent
○ good ○ mid ○ new ○ poor ○ -

Question 3C One-level Tree. (Continued) [6 marks]

Show your working below (if you wish):

Question 3D Full Decision Tree. [6 marks]



What is node 0?

- ◯ Age ◯ Condition ◯ Rooms ◯ High
- ◯ Low ◯ Low/High ◯ -

What is node 1?

- ◯ Age ◯ Condition ◯ Rooms ◯ High
- ◯ Low ◯ Low/High ◯ -

What is node 2?

- ◯ Age ◯ Condition ◯ Rooms ◯ High
- ◯ Low ◯ Low/High ◯ -

What is node 3?

- ◯ Age ◯ Condition ◯ Rooms ◯ High
- ◯ Low ◯ Low/High ◯ -

What is node 4?

- ◯ Age ◯ Condition ◯ Rooms ◯ High
- ◯ Low ◯ Low/High ◯ -

What is node 5?

- ◯ Age ◯ Condition ◯ Rooms ◯ High
- ◯ Low ◯ Low/High ◯ -

What is node 6?

- ◯ Age ◯ Condition ◯ Rooms ◯ High
- ◯ Low ◯ Low/High ◯ -

Are there any nodes from 7 onwards?

- ◯ Yes ◯ No

Question 3D Full Decision Tree. (Continued) [6 marks]

What is edge a?

◯ 2	◯ 3	◯ aged	◯ excellent
◯ mid	◯ new	◯ poor	◯ -

What is edge b?

◯ 2	◯ 3	◯ aged	◯ excellent
◯ mid	◯ new	◯ poor	◯ -

What is edge c?

◯ 2	◯ 3	◯ aged	◯ excellent
◯ mid	◯ new	◯ poor	◯ -

What is edge d?

◯ 2	◯ 3	◯ aged	◯ excellent
◯ mid	◯ new	◯ poor	◯ -

What is edge e?

◯ 2	◯ 3	◯ aged	◯ excellent
◯ mid	◯ new	◯ poor	◯ -

What is edge f?

◯ 2	◯ 3	◯ aged	◯ excellent
◯ mid	◯ new	◯ poor	◯ -

Question 3D Full Decision Tree. (Continued) [6 marks]

Show your working below (if you wish):

Question 3E Precision, Recall, F1. [5 marks]

Number of True Positives (TP)?

◯ 0 ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ >5

Number of False Positives (FP)?

◯ 0 ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ >5

Number of True Negatives(TN)?

◯ 0 ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ >5

Number of False Negatives (FN)?

◯ 0 ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ >5

Precision?

◯ $0.0 \leq \text{Precision} < 0.2$ ◯ $0.2 \leq \text{Precision} < 0.4$
◯ $0.4 \leq \text{Precision} < 0.6$ ◯ $0.6 \leq \text{Precision} < 0.8$
◯ $0.8 \leq \text{Precision} \leq 1.0$

Recall?

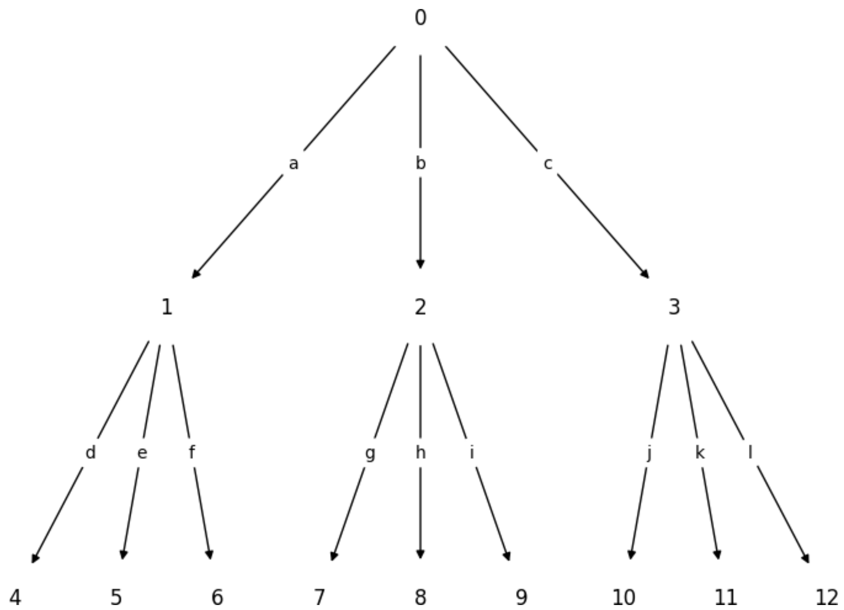
◯ $0.0 \leq \text{Recall} < 0.2$ ◯ $0.2 \leq \text{Recall} < 0.4$
◯ $0.4 \leq \text{Recall} < 0.6$ ◯ $0.6 \leq \text{Recall} < 0.8$
◯ $0.8 \leq \text{Recall} \leq 1.0$

F1 Score?

◯ $0.0 \leq \text{F1 Score} < 0.2$ ◯ $0.2 \leq \text{F1 Score} < 0.4$
◯ $0.4 \leq \text{F1 Score} < 0.6$ ◯ $0.6 \leq \text{F1 Score} < 0.8$
◯ $0.8 \leq \text{F1 Score} \leq 1.0$

Show your working below (if you wish):

Question 3F Min-Sample DT. [3 marks]



What is node 0?

◯ Age ◯ Condition ◯ Location ◯ Rooms
◯ High ◯ Low ◯ Low/High ◯ -

What is node 1?

◯ Age ◯ Condition ◯ Location ◯ Rooms
◯ High ◯ Low ◯ Low/High ◯ -

What is node 2?

◯ Age ◯ Condition ◯ Location ◯ Rooms
◯ High ◯ Low ◯ Low/High ◯ -

What is node 3?

◯ Age ◯ Condition ◯ Location ◯ Rooms
◯ High ◯ Low ◯ Low/High ◯ -

What is node 4?

◯ Age ◯ Condition ◯ Location ◯ Rooms
◯ High ◯ Low ◯ Low/High ◯ -

What is node 5?

◯ Age ◯ Condition ◯ Location ◯ Rooms
◯ High ◯ Low ◯ Low/High ◯ -

What is node 6?

◯ Age ◯ Condition ◯ Location ◯ Rooms
◯ High ◯ Low ◯ Low/High ◯ -

Are there any nodes from 7 onwards?

◯ Yes ◯ No

Question 3F Min-Sample DT. (Continued) [3 marks]

What is edge a?

◯ 2 ◯ 3 ◯ aged ◯ bad ◯ excellent
◯ good ◯ mid ◯ new ◯ poor ◯ -

What is edge b?

◯ 2 ◯ 3 ◯ aged ◯ bad ◯ excellent
◯ good ◯ mid ◯ new ◯ poor ◯ -

What is edge c?

◯ 2 ◯ 3 ◯ aged ◯ bad ◯ excellent
◯ good ◯ mid ◯ new ◯ poor ◯ -

What is edge d?

◯ 2 ◯ 3 ◯ aged ◯ bad ◯ excellent
◯ good ◯ mid ◯ new ◯ poor ◯ -

What is edge e?

◯ 2 ◯ 3 ◯ aged ◯ bad ◯ excellent
◯ good ◯ mid ◯ new ◯ poor ◯ -

What is edge f?

◯ 2 ◯ 3 ◯ aged ◯ bad ◯ excellent
◯ good ◯ mid ◯ new ◯ poor ◯ -

Show your working below (if you wish):

Question 4A Game Tree. [6 marks]

What is the minimum number of variables required?

◯ 1 ◯ 2 ◯ 3 ◯ 7 ◯ 8 ◯ Other

How many nodes (not including the leaves) are there in the game tree?

◯ 1	◯ 2	◯ 3	◯ 4	◯ 5	◯ 6	◯ 7
◯ 8	◯ 9	◯ 10	◯ 11	◯ 12	◯ 13	◯ 14
◯ 15	◯ 16	◯ 17	◯ 18	◯ 19	◯ 20	◯ >20

What is the number of leaves where the first player wins?

◯ 0 ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6 ◯ >6

What is the number of leaves where the first player loses?

◯ 0 ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6 ◯ >6

What is the total number of edges in the game tree?

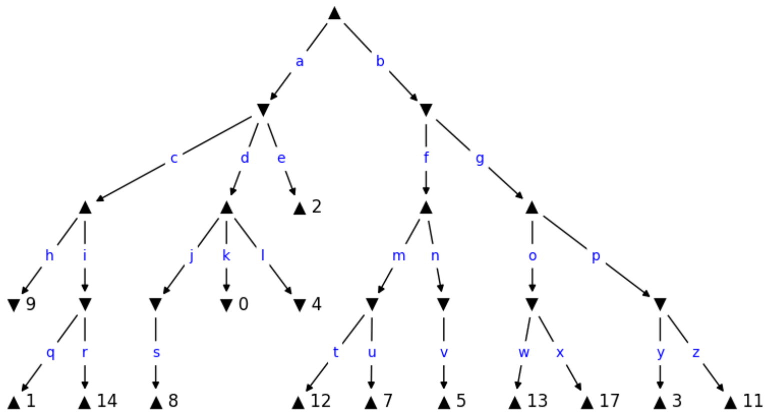
◯ 1	◯ 2	◯ 3	◯ 4	◯ 5	◯ 6	◯ 7
◯ 8	◯ 9	◯ 10	◯ 11	◯ 12	◯ 13	◯ 14
◯ 15	◯ 16	◯ 17	◯ 18	◯ 19	◯ 20	◯ >20

Which player do we expect to win?

◯ First player ◯ Second player ◯ It depends ◯ Draw

Show your working below (if you wish):

Question 4B Alpha-beta left-to-right. [5 marks]



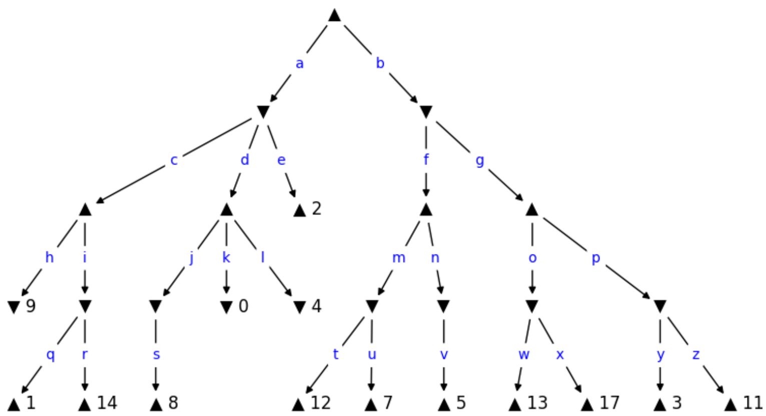
What is the root value?

○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 7
○ 8 ○ 9 ○ 11 ○ 12 ○ 13 ○ 14 ○ 17

Which of the following link(s) are pruned? Shade all that is/are true.

○ a ○ b ○ c ○ d ○ e ○ f
○ g ○ h ○ i ○ j ○ k ○ l
○ m ○ n ○ o ○ p ○ q ○ r
○ s ○ t ○ u ○ v ○ w ○ x
○ y ○ z

Question 4C Alpha-beta right-to-left. [4 marks]



Which of the following link(s) are pruned? Shade all that is/are true.

○ a ○ b ○ c ○ d ○ e ○ f
○ g ○ h ○ i ○ j ○ k ○ l
○ m ○ n ○ o ○ p ○ q ○ r
○ s ○ t ○ u ○ v ○ w ○ x
○ y ○ z

Question 5 3 most important lessons [3 marks]

This page is intentionally left blank.

It may be used as scratch paper.

— END OF ANSWER SHEET —

Question 1A State Representation [2 marks]

This is somewhat of a trick question. We just need to keep track of one integer that is the current location of the frog.

There are 4 possible actions: either 1 step or k steps forward. It is not wrong for the frog to go backwards, but it is not necessary for this question and this will create repeated states and change the answers to the later parts.

[+1] for the correct state representation.
[+1] for stating the 4 state transitions (actions).

Question 1B Initial and goal states [2 marks]

Initial state: $p = 0$, where p is the position of the frog

Goal state: $p > n$.

[+1] for initial state.
[+1] for goal state.

Note that if (A) is blank or wrong, this question will automatically be zero marks.

Question 1C Which of the following statement(s) is/are true? [3 marks]

- ○ The search tree is finite.
- ● There are possibly many repeated states.
- ● There are possibly multiple goal states.
- ○ We can always find an answer (valid sequence of moves) if we employ the right search algorithm.
- ○ None of the above.

Question 1D Best search algorithm. Explain. [2 marks]

BFS, because we need the minimal number of steps.
Not DFS, because no guarantee that it will be minimal.
Not DLS, because again, no guarantee that it will be minimal.
Not IDS, because while it works, We are paying extra for no good reason.

[+1] for BFS.
[+1] for saying something sensible about 2 of the algorithms.
Note that the correct answer here depends on depends on Parts (A) to (C). Saying that none of them are any good because they cannot necessarily terminate will give a small amount of credit (+1).

Question 1E State Representation for Variant #1 [2 marks]

Now we need to track 2 integers: (i) position of the frog; and (ii) number of moves left.

There are now $2k$ possible actions: up to k steps backwards or forwards

[+1] for adding the number of moves left.
[+1] for updating list of possible actions.

The right approach here is actually somewhat dependent on the search algorithm to be used. There are 2 obvious approaches: (i) track number of moves and do DFS, or (ii) do not track the number of moves and do DLS. For the former, the failure to track the number of moves will result in zero for this part and most of the subsequent parts.

Question 1F Initial and goal states for Variant #1 [2 marks]

Initial state: $p = 0$, moves $= 0$ Goal state: $p > n$, moves $= m$

[+1] for initial state.
[+1] for goal state.

Question 1G Invariant for Variant #1 [2 marks]

moves $\leq m$ `seq[p]` $\neq$ `"_"`

Question 1H Which of the following statement(s) is/are true for Variant #1? [3 marks]

- ● The search tree is finite.
- ● There are possibly many repeated states.
- ● There are possibly multiple goal states.
- ○ We can always find an answer (valid sequence of moves) if we employ the right search algorithm.
- ○ None of the above.

Question 1I Best search algorithm for Variant #1. Explain. [2 marks]

DLS, because we know exactly how many moves we need.
BFS, also works.
Not DFS, might not find a solution.
Not IDS, because while it works, We are paying extra for no good reason.

[+1] for DLS/BFS.
[+1] for saying something sensible.
Note that the correct answer here depends on depends on Parts (E) to (G).

Question 1J To Graph-Search or not? That is the question. [2 marks]

It depends! If we track the number of moves, then we can do Graph Search, but if we do not tack the number of moves and do DLS then we might actually <u>need</u> the repeated states!! The moral of the story here is that repeated states are not always bad and removing them might be harmful if people don't know what they are doing. But it really depends on the answer in Part (I).

Question 1K State Representation for Variant #2 [2 marks]

Now we need to track 3 integers: (i) position of the frog; (ii) number of moves left and (iii) amount of energy left E.

Actions remain unchanged.

[+1] for adding the number of moves left.
[+1] for updating list of possible actions.

Question 1L Which of the following statement(s) is/are true for Variant #2? [3 marks]

- ● The search tree is finite.
- ● There are possibly many repeated states.
- ● There are possibly multiple goal states.
- ○ We can always find an answer (valid sequence of moves) if we employ the right search algorithm.
- ○ None of the above.

The answer here depends on Parts (K).

Question 1M Best search algorithm for Variant #2. Explain. [3 marks]

Basically will need to use uniform cost search based on the energy consumption (with pruning at E) since we need to minimize energy consumption. The search tree is finite, so we will definitely be able to determine if there's not solution.

[+1] for UCS, based on E instead of m.
[+1] for saying need to cut off at E.
[+1] for saying that we can always determine if there's a solution because search tree is finite.
Note that the correct answer here depends on depends on Part (K).

Question 2A Is h_1 admissible? Explain. [2 marks]

○ Admissible ● Not admissible

Consider the case where the position of the character is at (0,0), box 1 at (1,0), box 2 at (2,0), goal 1 at (2,0), and goal 2 at (3,0). In this case, $h_1 = 2 > h^* = 1$.

Question 2B Is h_2 admissible? Explain. [2 marks]

● Admissible ○ Not admissible

Consider the case of Sokoban with N boxes and goals. In the goal state s_g, all boxes are in their respective goal positions, thus $h_2(s_g) = \frac{0}{N} = 0 \leq h^*(s_g)$.

For non-goal states, the character can push $M \leq N$ misplaced boxes to their goal positions. The minimum number of move required to move M boxes to their goal positions is 1. For M misplaces boxes, the heuristic gives $h_2(s) = \frac{M}{N} \leq 1 \leq h^*(s)$.

Question 2C Is h_A admissible? Explain. [2 marks]

● Admissible ○ Not admissible

Consider a case of Sokoban with N boxes and goals. In the goal state s_g, all boxes are in their respective goal positions, thus $h_A(s_g) = 0 \leq h^*(s_g)$.

In contrast to numbered Sokoban, at each step, the character can push at most 1 box to the goal. Thus, when there are $M \leq N$ number of misplaces boxes, at least it will take M number of steps to put all boxes to their respective goals. Thus, $\forall_{s \in S} h_A(s) \leq h^*(s)$, where S is a set of states.

Question 2D h_B admissible and consistent? Explain. [6 marks]

○ Admissible ● Not admissible

Consider the following case: $B = \{(1,0),(2,0)\}, G = \{(1,0),(2,0)\}$. In this case, $h_B = 1 > h^* = 0$.

○ Consistent ● Not consistent

We know that consistency implies admissibility. Thus, if a heuristic is not admissible, it is also not consistent (by the contrapositive).

Question 2E Is h_C admissible and consistent? Explain. [6 marks]

● Admissible ○ Not admissible

Consider a relaxed game of Sokoban where the character can jump to any position with 0 cost and need to only push one box which has the maximum closest distance to its goal position. In this relaxed problem, the number of steps required to solve the game is exactly the heuristic cost h_C. Thus, h_C is admissible.

● Consistent ○ Not consistent

h_C is the exact cost for the relaxed game of Sokoban as described above, thus h_C is consistent.

Question 2F h_D admissible and consistent? Explain. [6 marks]

● Admissible ○ Not admissible

To solve the game, the character minimally needs to move (zero the gap) to a box (which may or may not be closest), then to a goal position (which also may or may not be the closest). Thus, $\forall_{s \in S} h_D(s) \leq h^*(s)$, where S is a set of states.

○ Consistent ● Not consistent

Consider the following case. We start at a state u: $p = (0,1), G = \{(1,0),(0,4)\}, B = \{(1,0),(0,2)\}$ which has $h_D(u) = 2$. Suppose that we choose an action a to move right, then it will result in a state v: $p = (1,1), G = \{(1,0),(0,4)\}, B = \{(1,0),(0,2)\}$ with $h_D(v) = 0$. Since $c(u,a,v) = 1$, then $h_D(u) = 2 > c(u,a,v) + h_D(v) = 1 + 0$.

Question 2G Dominant relationships between the heuristics. [4 marks]

Which heuristic(s) does h_A dominate? Shade all that apply/ies.

○ h_B ○ h_C ○ h_D ● None

h_B is not admissible, thus there is no dominant relationship. h_C and h_D are sometimes larger and sometimes smaller, so there is no dominant relationship as well.

Which heuristic(s) does h_B dominates Shade all that apply/ies.

○ h_A ○ h_C ○ h_D ● None

h_B is not admissible, thus there is no dominant relationship with other heuristics.

Which heuristic(s) does h_C dominate? Shade all that apply/ies.

○ h_A ○ h_B ○ h_D ● None

h_B is not admissible, thus there is no dominant relationship. h_A and h_D are sometimes larger and sometimes smaller, so there is no dominant relationship as well.

Which heuristic(s) does h_D dominate? Shade all that apply/ies.

○ h_A ○ h_B ○ h_C ● None

h_B is not admissible, thus there is no dominant relationship. h_A and h_C are sometimes larger and sometimes smaller, so there is no dominant relationship as well.

Question 3A The price of a 2-room house? [2 marks]

○ High ● Low ○ Undefined

This question tests the student's ability to reason about a decision tree.

According to the given decision tree, if the age of a house is either aged or mid, then the price is low, irrespective of the number of rooms. On the other hand, if the age is new, then the price of a 2-rooms house is low.

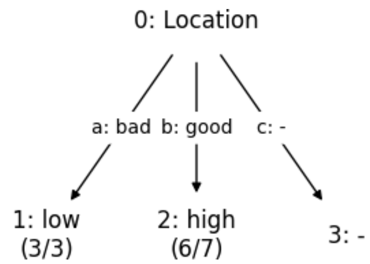
Question 3B What is the entropy? [2 marks]

- ○ $0.0 \leq \text{Entropy} < 0.2$
- ○ $0.2 \leq \text{Entropy} < 0.4$
- ○ $0.4 \leq \text{Entropy} < 0.6$
- ○ $0.6 \leq \text{Entropy} < 0.8$
- ● $0.8 \leq \text{Entropy} \leq 1.0$

Show your working below (if you wish):

$$\begin{aligned} Entropy &= I(\frac{6}{10}, \frac{4}{10}) \\ &= -\frac{6}{10}\log_2(\frac{6}{10}) - \frac{4}{10}\log_2(\frac{4}{10}) \\ &= 0.971 \end{aligned}$$

Question 3C One-level Tree. [6 marks]



Note: (X/Y) in the leaf nodes represent the number of samples belonging to the majority class over the total number of samples across all classes. To illustrate, a leaf node labeled as "high (6/7)" indicates that out of 7 samples, 6 of them belong to the "high" class.

What is node 0?

○ Age ○ Condition ● Location ○ Rooms
○ High ○ Low ○ Low/High ○ -

What is node 1?

○ Age ○ Condition ○ Location ○ Rooms
○ High ● Low ○ Low/High ○ -

What is node 2?

○ Age ○ Condition ○ Location ○ Rooms
● High ○ Low ○ Low/High ○ -

What is node 3?

○ Age ○ Condition ○ Location ○ Rooms
○ High ○ Low ○ Low/High ● -

What is edge a?

○ 2 ○ 3 ○ aged ● bad ○ excellent
○ good ○ mid ○ new ○ poor ○ -

What is edge b?

○ 2 ○ 3 ○ aged ○ bad ○ excellent
● good ○ mid ○ new ○ poor ○ -

What is edge c?

○ 2 ○ 3 ○ aged ○ bad ○ excellent
○ good ○ mid ○ new ○ poor ● -

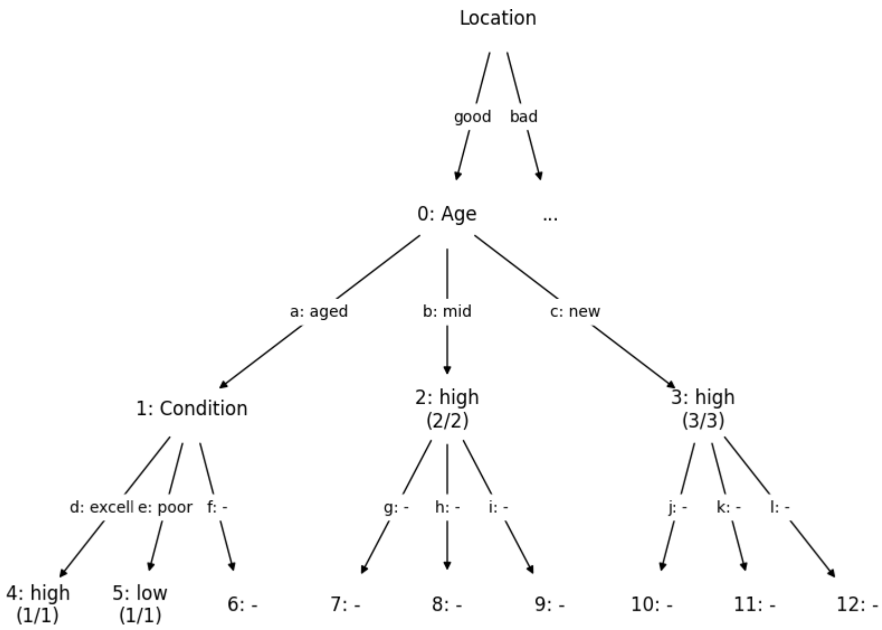
Question 3C One-level Tree. (Continued) [6 marks]

Show your working below (if you wish):

$$
\begin{aligned}
Remainder(Location) &= \frac{7}{10}\big(I(\frac{6}{7},\frac{1}{7})\big) + \frac{3}{10}\big(I(\frac{3}{3})\big) \\
&= \frac{7}{10}\big(-\frac{6}{7}\log_2(\frac{6}{7}) - \frac{1}{7}\log_2(\frac{1}{7})\big) + \frac{3}{10}\big(-\frac{3}{3}\log_2(\frac{3}{3})\big) \\
&= \frac{7}{10}\big(0.592\big) + \frac{3}{10}\big(0.000\big) \\
&= 0.414 + 0.000 \\
&= 0.414 \\
Remainder(Condition) &= \frac{3}{10}\big(I(\frac{3}{3})\big) + \frac{7}{10}\big(I(\frac{4}{7},\frac{3}{7})\big) \\
&= \frac{3}{10}\big(-\frac{3}{3}\log_2(\frac{3}{3})\big) + \frac{7}{10}\big(-\frac{4}{7}\log_2(\frac{4}{7}) - \frac{3}{7}\log_2(\frac{3}{7})\big) \\
&= \frac{3}{10}\big(0.000\big) + \frac{7}{10}\big(0.985\big) \\
&= 0.000 + 0.690 \\
&= 0.690 \\
Remainder(Rooms) &= \frac{5}{10}\big(I(\frac{4}{5},\frac{1}{5})\big) + \frac{5}{10}\big(I(\frac{3}{5},\frac{2}{5})\big) \\
&= \frac{5}{10}\big(-\frac{4}{5}\log_2(\frac{4}{5}) - \frac{1}{5}\log_2(\frac{1}{5})\big) + \frac{5}{10}\big(-\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5})\big) \\
&= \frac{5}{10}\big(0.722\big) + \frac{5}{10}\big(0.971\big) \\
&= 0.361 + 0.485 \\
&= 0.846 \\
Remainder(Age) &= \frac{2}{10}\big(I(\frac{1}{2},\frac{1}{2})\big) + \frac{3}{10}\big(I(\frac{2}{3},\frac{1}{3})\big) + \frac{5}{10}\big(I(\frac{3}{5},\frac{2}{5})\big) \\
&= \frac{2}{10}\big(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})\big) + \frac{3}{10}\big(-\frac{2}{3}\log_2(\frac{2}{3}) - \frac{1}{3}\log_2(\frac{1}{3})\big) \\
&\quad + \frac{5}{10}\big(-\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5})\big) \\
&= \frac{2}{10}\big(1.000\big) + \frac{3}{10}\big(0.918\big) + \frac{5}{10}\big(0.971\big) \\
&= 0.200 + 0.275 + 0.485 \\
&= 0.96
\end{aligned}
$$

We split based on "Location" as it results in the smallest remaining entropy.

Question 3D Full Decision Tree. [6 marks]



What is node 0?

● Age ○ Condition ○ Rooms ○ High
○ Low ○ Low/High ○ -

What is node 1?

○ Age ● Condition ○ Rooms ○ High
○ Low ○ Low/High ○ -

What is node 2?

○ Age ○ Condition ○ Rooms ● High
○ Low ○ Low/High ○ -

What is node 3?

○ Age ○ Condition ○ Rooms ● High
○ Low ○ Low/High ○ -

What is node 4?

○ Age ○ Condition ○ Rooms ● High
○ Low ○ Low/High ○ -

What is node 5?

○ Age ○ Condition ○ Rooms ○ High
● Low ○ Low/High ○ -

What is node 6?

○ Age ○ Condition ○ Rooms ○ High
○ Low ○ Low/High ● -

Are there any nodes from 7 onwards?

○ Yes ● No

Question 3D Full Decision Tree. (Continued) [6 marks]

What is edge a?

○ 2 ○ 3 ● aged ○ excellent
○ mid ○ new ○ poor ○ -

What is edge b?

○ 2 ○ 3 ○ aged ○ excellent
● mid ○ new ○ poor ○ -

What is edge c?

○ 2 ○ 3 ○ aged ○ excellent
○ mid ● new ○ poor ○ -

What is edge d?

○ 2 ○ 3 ○ aged ● excellent
○ mid ○ new ○ poor ○ -

What is edge e?

○ 2 ○ 3 ○ aged ○ excellent
○ mid ○ new ● poor ○ -

What is edge f?

○ 2 ○ 3 ○ aged ○ excellent
○ mid ○ new ○ poor ● -

Question 3D Full Decision Tree. (Continued) [6 marks]

Show your working below (if you wish):

$$
\begin{aligned}
Remainder(Age) &= \frac{2}{7}\big(I(\frac{1}{2},\frac{1}{2})\big) + \frac{2}{7}\big(I(\frac{2}{2})\big) + \frac{3}{7}\big(I(\frac{3}{3})\big) \\
&= \frac{2}{7}\big(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})\big) + \frac{2}{7}\big(-\frac{2}{2}\log_2(\frac{2}{2})\big) + \frac{3}{7}\big(-\frac{3}{3}\log_2(\frac{3}{3})\big) \\
&= \frac{2}{7}(1.000) + \frac{2}{7}(0.000) + \frac{3}{7}(0.000) \\
&= 0.286 + 0.000 + 0.000 \\
&= 0.286 \\
Remainder(Condition) &= \frac{3}{7}\big(I(\frac{3}{3})\big) + \frac{4}{7}\big(I(\frac{1}{4},\frac{3}{4})\big) \\
&= \frac{3}{7}\big(-\frac{3}{3}\log_2(\frac{3}{3})\big) + \frac{4}{7}\big(-\frac{1}{4}\log_2(\frac{1}{4}) - \frac{3}{4}\log_2(\frac{3}{4})\big) \\
&= \frac{3}{7}(0.000) + \frac{4}{7}(0.811) \\
&= 0.000 + 0.464 \\
&= 0.464 \\
Remainder(Rooms) &= \frac{5}{7}\big(I(\frac{4}{5},\frac{1}{5})\big) + \frac{2}{7}\big(I(\frac{2}{2})\big) \\
&= \frac{5}{7}\big(-\frac{4}{5}\log_2(\frac{4}{5}) - \frac{1}{5}\log_2(\frac{1}{5})\big) + \frac{2}{7}\big(-\frac{2}{2}\log_2(\frac{2}{2})\big) \\
&= \frac{5}{7}(0.722) + \frac{2}{7}(0.000) \\
&= 0.516 + 0.000 \\
&= 0.516
\end{aligned}
$$

We split based on "Age" as it results in the smallest remaining entropy.

If Age = "aged", then we split again:

	Rooms	Condition	Price
0	2	excellent	High
1	2	poor	Low

Based on the table, it's clear that we should split based on "Condition".

Question 3E Precision, Recall, F1. [5 marks]

Number of True Positives (TP)?

○ 0 ● 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ >5

Number of False Positives (FP)?

○ 0 ○ 1 ● 2 ○ 3 ○ 4 ○ 5 ○ >5

Number of True Negatives(TN)?

○ 0 ○ 1 ● 2 ○ 3 ○ 4 ○ 5 ○ >5

Number of False Negatives (FN)?

○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ● 5 ○ >5

Precision?

○ $0.0 \leq \text{Precision} < 0.2$ ● $0.2 \leq \text{Precision} < 0.4$
○ $0.4 \leq \text{Precision} < 0.6$ ○ $0.6 \leq \text{Precision} < 0.8$
○ $0.8 \leq \text{Precision} \leq 1.0$

Recall?

● $0.0 \leq \text{Recall} < 0.2$ ○ $0.2 \leq \text{Recall} < 0.4$
○ $0.4 \leq \text{Recall} < 0.6$ ○ $0.6 \leq \text{Recall} < 0.8$
○ $0.8 \leq \text{Recall} \leq 1.0$

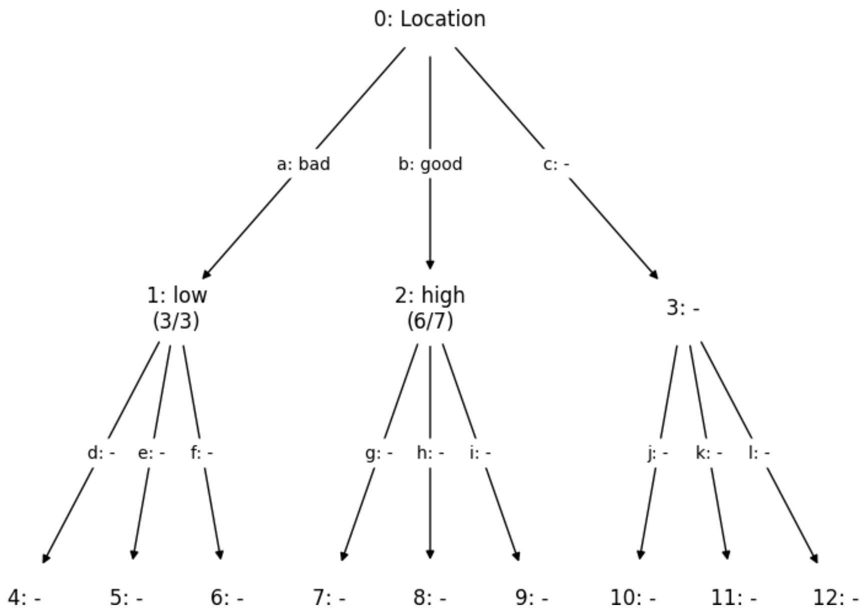
F1 Score?

○ $0.0 \leq \text{F1 Score} < 0.2$ ● $0.2 \leq \text{F1 Score} < 0.4$
○ $0.4 \leq \text{F1 Score} < 0.6$ ○ $0.6 \leq \text{F1 Score} < 0.8$
○ $0.8 \leq \text{F1 Score} \leq 1.0$

Show your working below (if you wish):

$$
\begin{aligned}
Precision &= \frac{TP}{TP+FP} \\
&= \frac{1}{1+2} \\
&= 0.333 \\
Recall &= \frac{TP}{TP+FN} \\
&= \frac{1}{1+5} \\
&= 0.167 \\
F1 &= \frac{2}{\frac{1}{P}+\frac{1}{R}} \\
&= \frac{2}{\frac{1}{0.333}+\frac{1}{0.167}} \\
&= 0.222
\end{aligned}
$$

Question 3F Min-Sample DT. [3 marks]



What is node 0?

○ Age ○ Condition ● Location ○ Rooms
○ High ○ Low ○ Low/High ○ -

What is node 1?

○ Age ○ Condition ○ Location ○ Rooms
○ High ● Low ○ Low/High ○ -

What is node 2?

○ Age ○ Condition ○ Location ○ Rooms
● High ○ Low ○ Low/High ○ -

What is node 3?

○ Age ○ Condition ○ Location ○ Rooms
○ High ○ Low ○ Low/High ● -

What is node 4?

○ Age ○ Condition ○ Location ○ Rooms
○ High ○ Low ○ Low/High ● -

What is node 5?

○ Age ○ Condition ○ Location ○ Rooms
○ High ○ Low ○ Low/High ● -

What is node 6?

○ Age ○ Condition ○ Location ○ Rooms
○ High ○ Low ○ Low/High ● -

Are there any nodes from 7 onwards?

○ Yes ● No

Question 3F Min-Sample DT. (Continued) [3 marks]

What is edge a?

- ○ 2 ○ 3 ○ aged ● bad ○ excellent
- ○ good ○ mid ○ new ○ poor ○ -

What is edge b?

- ○ 2 ○ 3 ○ aged ○ bad ○ excellent
- ● good ○ mid ○ new ○ poor ○ -

What is edge c?

- ○ 2 ○ 3 ○ aged ○ bad ○ excellent
- ○ good ○ mid ○ new ○ poor ● -

What is edge d?

- ○ 2 ○ 3 ○ aged ○ bad ○ excellent
- ○ good ○ mid ○ new ○ poor ● -

What is edge e?

- ○ 2 ○ 3 ○ aged ○ bad ○ excellent
- ○ good ○ mid ○ new ○ poor ● -

What is edge f?

- ○ 2 ○ 3 ○ aged ○ bad ○ excellent
- ○ good ○ mid ○ new ○ poor ● -

Show your working below (if you wish):

Question 4A Game Tree. [6 marks]

What is the minimum number of variables required?

● 1 ○ 2 ○ 3 ○ 7 ○ 8 ○ Other

How many nodes (not including the leaves) are there in the game tree?

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7
○ 8 ● 9 ○ 10 ○ 11 ○ 12 ○ 13 ○ 14
○ 15 ○ 16 ○ 17 ○ 18 ○ 19 ○ 20 ○ >20

What is the number of leaves where the first player wins?

○ 0 ○ 1 ● 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ >6

What is the number of leaves where the first player loses?

○ 0 ● 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ >6

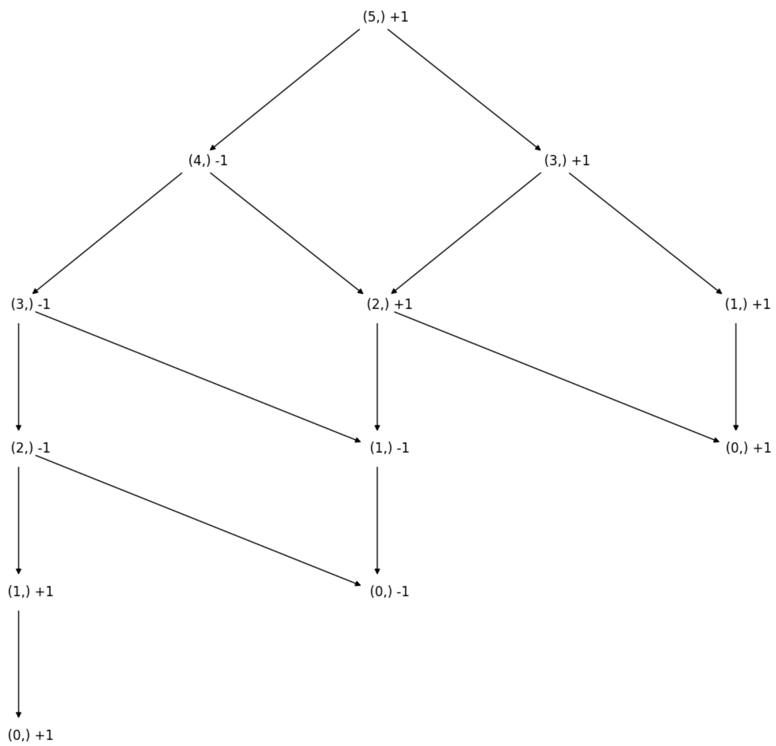
What is the total number of edges in the game tree?

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7
○ 8 ○ 9 ○ 10 ○ 11 ○ 12 ○ 13 ○ 14
● 15 ○ 16 ○ 17 ○ 18 ○ 19 ○ 20 ○ >20

Which player do we expect to win?

● First player ○ Second player ○ It depends ○ Draw

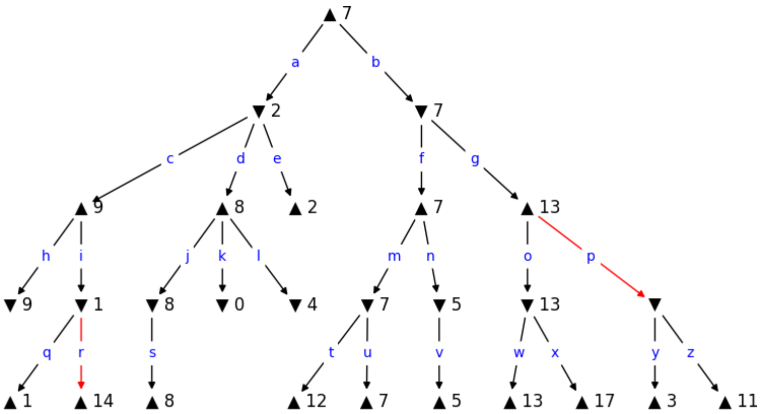
Show your working below (if you wish):



This question is here to test that the student knows how to draw a game tree and solve it correctly. This question is meant to be easy.

This is a Nim game where the starting state is the gap between two pegs.

Question 4B Alpha-beta left-to-right. [5 marks]



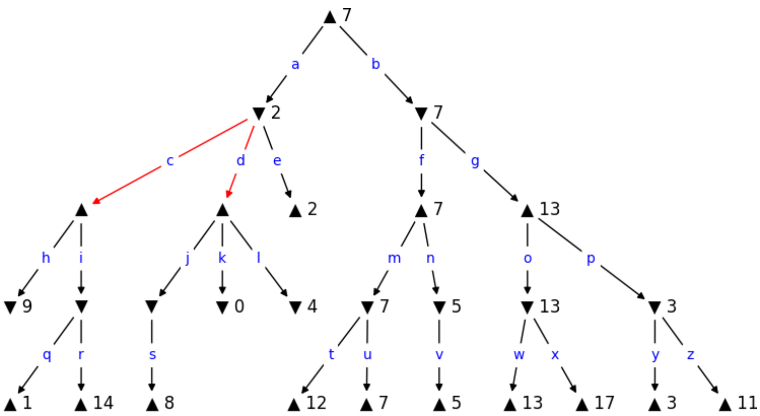
What is the root value?

○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ● 7
○ 8 ○ 9 ○ 11 ○ 12 ○ 13 ○ 14 ○ 17

Which of the following link(s) are pruned? Shade all that is/are true.

○ a ○ b ○ c ○ d ○ e ○ f
○ g ○ h ○ i ○ j ○ k ○ l
○ m ○ n ○ o ● p ○ q ● r
○ s ○ t ○ u ○ v ○ w ○ x
○ y ○ z

Question 4C Alpha-beta right-to-left. [4 marks]



Which of the following link(s) are pruned? Shade all that is/are true.

○ a ○ b ● c ● d ○ e ○ f
○ g ○ h ○ i ○ j ○ k ○ l
○ m ○ n ○ o ○ p ○ q ○ r
○ s ○ t ○ u ○ v ○ w ○ x
○ y ○ z

Question 5 3 most important lessons [3 marks]

There are no right answers here. Students will get credit for 3 well-explained learning points that are reasonable and justified. Student needs to put in *some* effort. Clearly if the student makes a patently false statement, marks will be deducted.