National University of Singapore

CS2109S—Introduction to AI and Machine Learning

MIDTERM ASSESSMENT

Semester 1, 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 **ELEVEN (11) pages** including this cover page.
3. Weightage of questions is given in square brackets. The maximum attainable score is 100.
4. This is a <u>**OPEN-SHEET**</u> assessment. You are allow one A4-sized double-sided cheat-sheet.
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: Number Addition Puzzles [38 marks]

Your young cousin Ben Bitdiddle comes to you with the following logic puzzle that he found online:

$$\begin{array}{ccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \\ \hline \end{array}$$

Your goal was to find a set of digits (between 0 and 9) that would make this relationship hold. Then you realize that there are infinitely many variants for this problem. For example, one could also be asked to solve for the following:

$$\begin{array}{ccccc} & S & A & V & E \\ + & M & O & R & E \\ \hline M & O & N & E & Y \\ \hline \end{array}$$

In general, such addition puzzles are set up as the sum of 2 numbers, each with up to n digits, and an answer of up to $n+1$ (or sometimes n) digits. The goal is to find a mapping for the m letters in the puzzle to the digits 0 to 9 so that the addition is valid. All the letters are unique digits (i.e., there is no repetition), so there can be at most 10 distinct letters in a puzzle ($m \leq 10$).

Instead of simply solving the original SEND-MORE-MONEY puzzle, you decided to do better, so that your cousin won't come bother you again. You will write him program that can solve **all** such puzzles.

Uninformed Search

A. Propose a state representation for this problem if we want to formulate it as a search problem and define the corresponding actions. [4 marks]

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

C. What are the initial and goal states for the problem under your proposed representation in Part (A)? [4 marks]

D. Which of the following statement(s) is/are true given your definition of the search problem in Parts (A) to (C)? Shade <u>all</u> that is/are true. [4 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 mapping) if we employ the right search algorithm.
- None of the above.

E. 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 use? Explain. [4 marks]

F. Should we have used GRAPH-SEARCH (See Appendix) instead? Explain. [2 marks]

G. You implemented the TREE-SEARCH algorithm in Part (E) and realized that it often runs quite slowly. What is the expected order of growth in time for your algorithm in terms of n, for a large n, where n is the maximum number of letters in the top 2 rows. [2 marks]

H. Suggest how we can to improve the TREE-SEARCH algorithm you chose in Part (E) to make it run faster. [3 marks]

Local Search

Your other cousin Alyssa P Hacker hears about you solving the problem with Search. She has also taken CS2109S and she suggests to you that you can also formulate a solution to the problem as a local search problem.

Recall that to formulate the solution as a local search problem, there are 3 steps:

1. Define a candidate solution
2. Define a transition function to generate new candidate solutions
3. Define a heuristic to evaluate the "goodness" of a candidate solution.

All 3 need to be designed in tandem because they have an impact on each other.

I. Based on the description of the addition puzzle above, define an initial candidate solution. [2 marks]

J. Define a reasonable heuristic function to evaluate the "goodness" of a candidate solution. Explain how this heuristic can also be used as a goal test to determine that we have a solution to the problem. [4 marks]

K. Define a reasonable transition function to generate new candidate solutions. [4 marks]

L. Which is the better approach: uninformed search or local search? Explain. [3 marks]

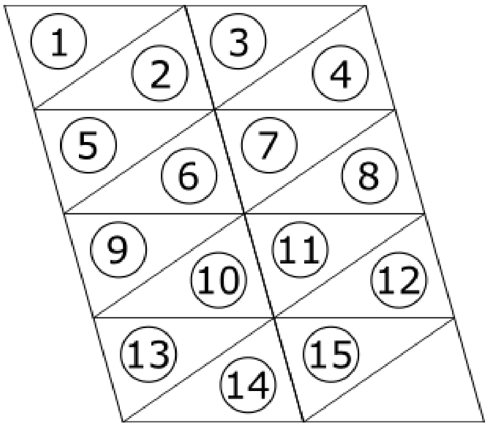
Question 2: Sliding Coins [23 marks]

Recall the Sliding Puzzle problem covered in the lecture, where tiles begins in some arrangement with one open space. In this question, we work with a variant of the Sliding Puzzle problem where the spaces are now triangular instead of square.

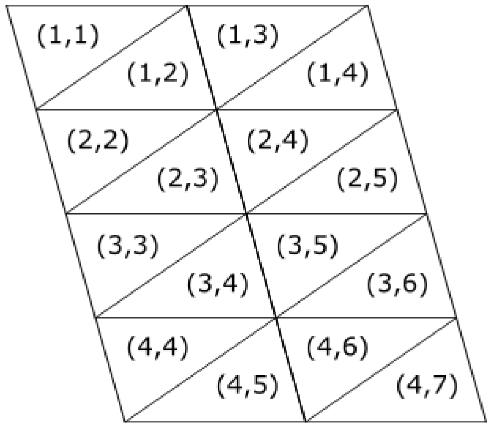
We have 16 triangular spaces, and fifteen coins numbered from 1 to 15. At each time-interval, a single numbered coin can move into a open space if the two spaces are adjacent (share edges). Each time-step has a cost of 1.

In this problem, the goal is to arrange the coins in sequence from 1 to 15 from left-to-right, top-to-bottom, and finally the leaving the final space at the bottom-right open. (see Fig. 1a)

In the goal state, coin 1 is in position (1, 1), and coin 15 is in position (4, 6).
Manhattan Distance (MD) is $|x_1 - x_2| + |y_1 - y_2|$, thus, $\text{MD}((1,1),(4,6)) = |1-4| + |1-6| = 8$.



(a) Goal state



(b) Unique coordinate scheme

Figure 1: Sliding Coin with triangular grid

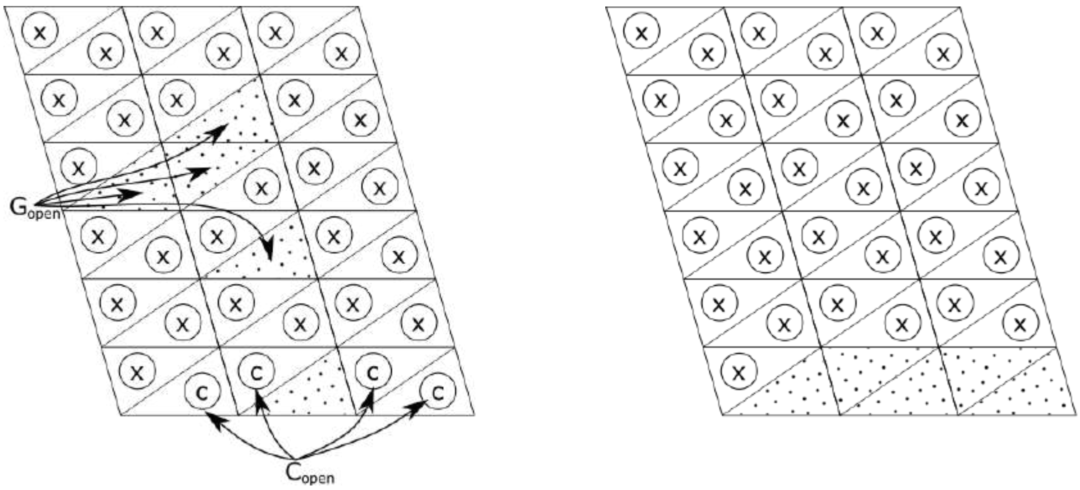
Your cousin Ben Bitdiddle immediately suggests using the same heuristics as discussed in the lectures for the original sliding puzzle problem.

A. **[Warm Up]** h_1 : number of misplaced coins
Is this heuristic admissible for the current problem? Explain. [2 marks]

B. h_2 : total Manhattan distance of each coin to their respective goal positions
Is this heuristic admissible for the current problem? Explain. [2 marks]

C. Consider a modified version of this problem where there are now k open spaces, and $n^2 - k$ coins, where $n^2 \gg k$, and n is even (Figure 2). Similar to before, at each time-step, we can only move a single coin into an open space if their edges are shared. In this k-blank variant, the **coins are not numbered, so ordering is irrelevant**.

What is the maximum branching factor for this modified version of the problem? Explain.
[2 marks]



(a) Unfinished state

(b) Goal state

Figure 2: Coin Sliding Puzzle where $n^2 = 36$ and $k = 5$

Ben Bitdiddle has come up with the following possible heuristics to be used for this new k-open spaces problem. 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.

Let $\mathrm{D}(a,b)$ be the actual path distance between locations a and b
and C_{open} be the set of positions of coins that are not yet in the goal positions
and G_{open} be the set of unoccupied/open goal positions of the coins.

D. Minimum of all distances of each coin to their farthest goal position. [4 marks]

$$h_A : \min_{c \in C_{open}} \{ \max_{g \in G_{open}} \{\mathrm{D}(c,g)\}\}$$

E. Sum of all distances of each coin to their closest goal position. [4 marks]

$$h_B : \sum_{c \in C_{open}} \min_{g \in G_{open}} \{D(c,g)\}$$

F. Sum of all distances of each coin to their farthest goal position. [4 marks]

$$h_C : \sum_{c \in C_{open}} \max_{g \in G_{open}} \{D(c,g)\}$$

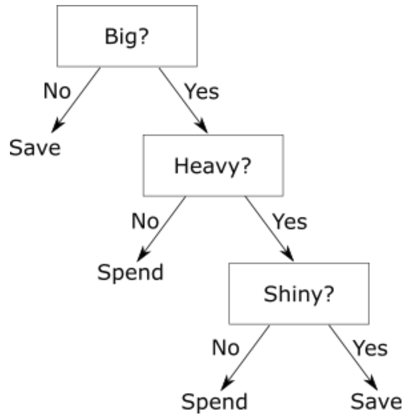
G. **[Dominance]** State and explain the dominant relationships between the 3 proposed heuristics in parts (D, E, F). [5 marks]

Question 3: To spend or to save? [21 marks]

Your young cousin Ben Bitdiddle has to decide whether to spend or save coins in his piggy bank! To make his decision, he has drawn up a decision tree based on the following table:

Big	Heavy	Shiny	**Decision**
No	No	No	Save
Yes	Yes	No	Spend
Yes	No	Yes	Spend
Yes	Yes	Yes	Save

(a)



(b)

Figure 3: Simple decision table and associated Decision Tree

A. **[Warm Up]** Ben is given a heavy and shiny coin. Based on the Decision Tree in Figure 3b, should Ben spend or save the coin? [2 marks]

B. It turns out that Ben's decision tree is not depth optimal! Help Ben optimize his decision tree by drawing a new decision tree that minimizes the depth. [2 marks]

In the end, Ben used a different set of features to decide whether to spend or save a coin. He recorded his decisions in the table below.

Coin	Rarity	Is Vintage	Condition	In Production	**Decision**
1	Rare	No	Rusty	Inactive	Spend
2	Rare	No	Rusty	Active	Save
3	Rare	Yes	Rusty	Active	Save
4	Rare	Yes	Rusty	Active	Save
5	Common	Yes	Rusty	Active	Spend
6	Common	Yes	Rusty	Active	Spend
7	Common	No	Rusty	Active	Spend
8	Common	No	Mint	Active	Spend
9	Rare	No	Mint	Active	Save
10	Rare	Yes	Mint	Inactive	Save

Table 1: Spend-or-save outcomes

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

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

C. What is the entropy of the decisions made in Table 1, rounded to 2 decimal places?
Hint: remember the log *is in base 2!* [2 marks]

D. Ben Bitdiddle is a busy child, and sometimes just wants to make a quick decision. To help him, you decided to create a one-level decision tree (only one split) using information gain.

Draw the decision tree, and clearly show how your chosen feature splits the data. Demonstrate why your decision tree is correct by using the appropriate calculations of different features. You may do your calculations at 2 decimal places precision. [6 marks]

E. Occasionally, Ben Bitdiddle *does* have time make rigorous decisions. In such cases, he would like to use all the information available to make his decision.

Construct the full decision tree based on the data (Table 1) using information gain to split the data. Show your working/reasoning for each node of the decision tree. Tie-breaking priority on tree construction is Rarity > Vintage > Condition > Production. [5 marks]

F. **[Min-Sample Pruning]** Suppose you want a Decision Tree where each leaf corresponds to at least 3 training data points. Derive this tree by pruning the tree you obtained in Part (E). Which coin(s) do you think are likely outlier(s)/error(s)? [4 marks]

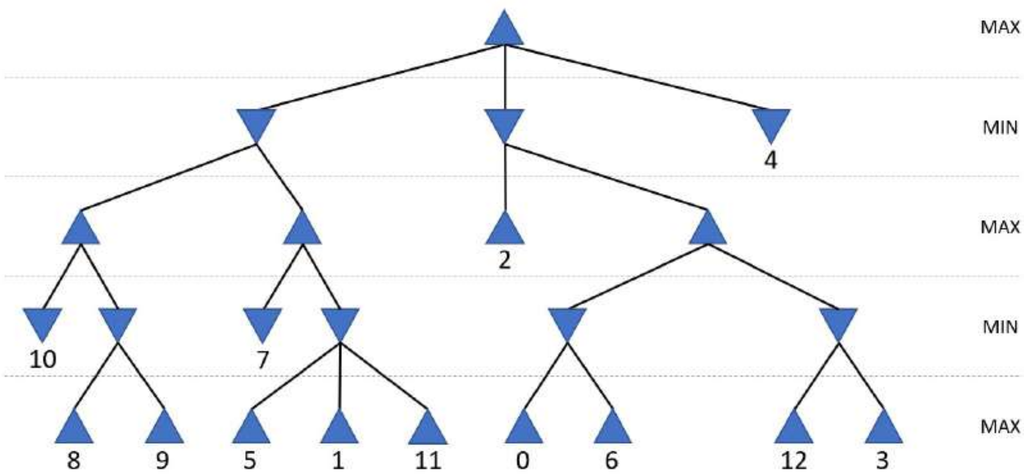
Question 4: Solving Games [15 marks]

A. **[Warm Up]** In lecture, we discussed the following game of Nim:

> Several piles of sticks are given. We represent the configuration of the piles by a **monotone** sequence of integers, such as (1,3,5). A player may remove, in one turn, any number of sticks from one pile. Thus, (1,3,5) would become (1,1,3) if the player were to remove 4 sticks from the last pile. The player who takes the last stick wins.

In this question, consider a modified game of Nim with 2 piles of sticks: one with 1 stick and one with 2 sticks. The player who takes the last stick **wins**. Draw the game tree and solve it. Which player do we expect to win? [6 marks]

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 left to right. Cross out the links that would be pruned by *alpha-beta*. 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. Cross out the links that would be pruned by *alpha-beta*. [4 marks]

Question 5: What have you learnt? [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 ← CHOOSE-ATTRIBUTE(*attributes*, *examples*)

tree ← 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 ← 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

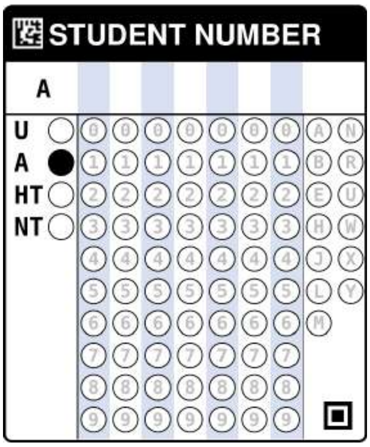
Midterm Assessment Answer Sheet

Semester 1, 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 **ELEVEN (11) 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.



For Examiner's Use Only

Question	Marks
Q1	/ 38
Q2	/ 23
Q3	/ 21
Q4	/ 15
Q5	/ 3
Total	/100

Question 1A State Representation [4 marks]

Question 1B Invariant [2 marks]

Question 1C Initial and goal states [4 marks]

Question 1D Which of the following statement(s) is/are true? [4 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 mapping) if we employ the right search algorithm.

◯ None of the above.

Question 1E Best search algorithm. [4 marks]

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

Question 1G Order of Growth (Time) [2 marks]

Question 1H Make it run faster! [3 marks]

Question 1I Initial candidate solution. [2 marks]

Question 1J Heuristic function. [4 marks]

Question 1K Transition function. [4 marks]

Question 1L Which is better: uninformed search or local search? [3 marks]

Question 2A Is h_1 admissible? [2 marks]

Question 2B Is h_2 admissible? [2 marks]

Question 2C Maximum Branching Factor [2 marks]

Question 2D h_A admissible and consistent? Explain. [4 marks]

○ Admissible ○ Not admissible

○ Consistent ○ Not consistent

Question 2E h_B admissible and consistent? Explain. [4 marks]

○ Admissible ○ Not admissible

○ Consistent ○ Not consistent

Question 2F h_C admissible and consistent? Explain. [4 marks]

○ Admissible ○ Not admissible

○ Consistent ○ Not consistent

Question 2G Dominant relationships between the heuristics. [5 marks]

Question 3A To save or to spend? [2 marks]

Question 3B Minimal-Depth Tree. [2 marks]

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

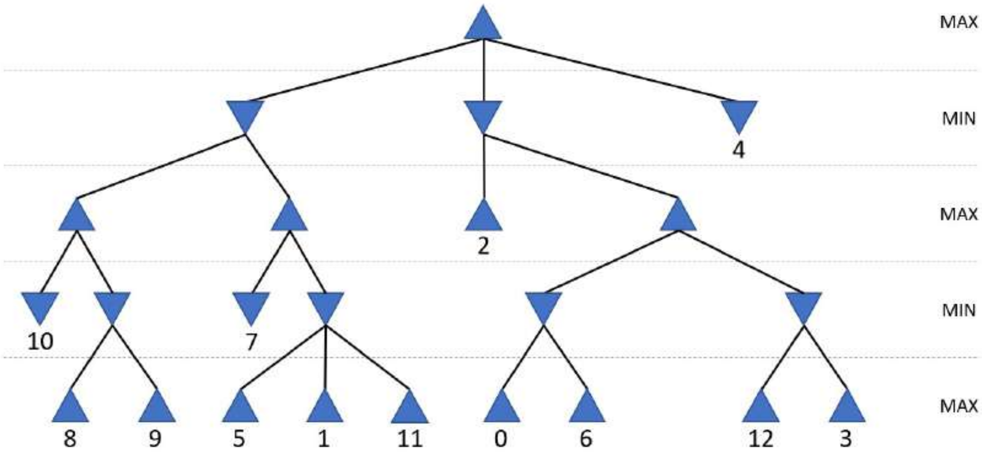
Question 3D One-level Tree. [6 marks]

Question 3E Full Decision Tree. [5 marks]

Question 3F Min-Sample DT [4 marks]

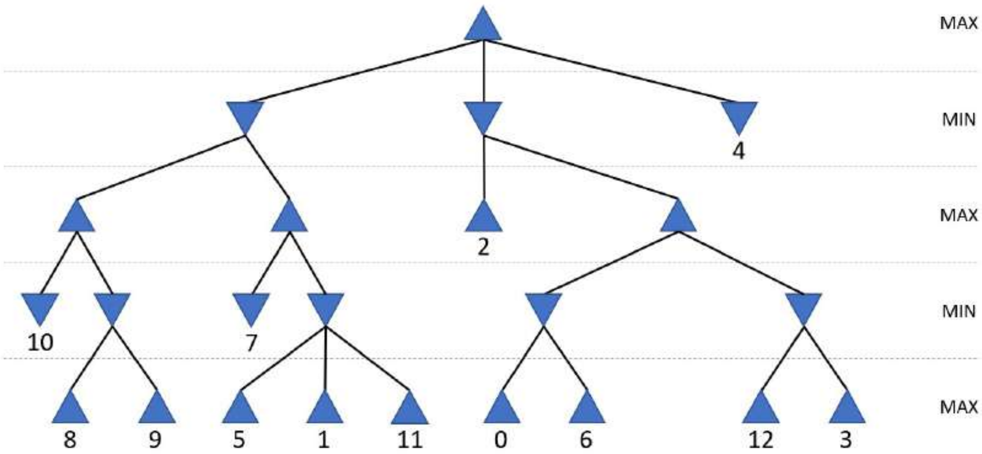
Question 4A Nim Game Tree. [6 marks]

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



Root value is:

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



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 [4 marks]

There are many possible solutions and credit will be award for any reasonable (and correct) solution. The most straightforward way to formulate this problem as uninformed search would be list all the variables (letters) in the puzzle. Let these variables be v_i, for $i = 1, \ldots, m$, $m \leq 10$.

The state of the search problem is a mapping of of $(v_1, \ldots, v_m)$ to digits. Some of the mappings could be null.

[+2] for the correct state representation.
[+2] for stating the correct actions with corresponding state transitions.

Question 1B Invariant [2 marks]

$v_i \neq v_j$, for all i, j where $i \neq j$ and v_i,v_j not null.

Question 1C Initial and goal states [4 marks]

Initial state: $(null, null, \cdots, null)$

Goal state: We won't know the goal state. If we knew the goal state, the problem is already solved! What we have is a goal test. The goal test has 2 parts:

1. There is a complete mapping, i.e., $v_i \neq$ null, for all i
2. The addition relationship is valid.

[+2] for initial state.
[+2] for goal test.

Question 1D Which of the following statement(s) is/are true? [4 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 mapping) if we employ the right search algorithm.
- ○ None of the above.

Question 1E Best search algorithm. [4 marks]

DFS, because it works and it does so with constant space.
Not BFS, because while it works, it takes more space than DFS.
Not DLS, because we don't know if we can find the solution if we limit the depth.
Not IDS, because while it works, We are paying extra for no good reason.

[+1] each for brief explanations for each algorithm.
Note that the correct answer here depends on depends on Parts (A) to (C).

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

No, there is no savings since we don't have repeated states. This question is here to test that the students understand the difference between TREE-SEARCH and GRAPH-SEARCH.

It is possible for the answer here to be yes, depending on Parts (A) to (C).

Question 1G Order of Growth (Time) [2 marks]

This is somewhat of a trick question. The search tree is bounded by $O(10^m)$. In fact, we know that there are at most 10! possibilities.

Question 1H Make it run faster! [3 marks]

Recall that a key technique to improve search is *pruning*. In this case, what we can do is to arrange to choose the letters at the lower decimal places to be used as the nodes earlier in the search. Then we also add an addition invariant/constraint that the first J columns from the right are all valid, i.e., when we added up the columns, there are no conflicts.

To make this more concrete, for the SEND-MORE-MONEY problem, we will choose letters for the nodes in the following order $\{$D,E,Y,N,R,O,S,M$\}$.

Question 1I Initial candidate solution. [2 marks]

There are a several ways to formulate this problem. The candidate solution needs only be a plausible solution to the problem.

For each v_i, assign at random one of the digits from 0 to 9, while ensuring that $v_i \neq v_j$, for all $i \neq j$ (i.e., that there are no repeated digits).

Question 1J Heuristic function. [4 marks]

There are 2 obvious candidates for heuristics:

1. Let $h = 0$ be the heuristic variable. Consider each of the $n+1$ columns from right to left. $h{+}{+}$ if the column does not add up. We take carry overs into account since we are considering the columsn from right to left.
2. Maintain a counter for each letter. Consider each of the $n+1$ columns from right to left. If the column does not add up, add one to the counters for all the letters in the column. We take carry overs into account since we are considering the columsn from right to left. The heuristic is the sum of all the counters.

We are counting conflicts, so if the heuristic is zero, we have our answer.

[+3] for a valid heuristic described clearly.
[+1] for highlighting that heuristic = 0 is a goal test.

Question 1K Transition function. [4 marks]

There are 2 cases:

1. If all the digits are used, i.e., $m = 10$. Then we just pick a random pair of letters that are involved in some conflict and swap them if we can reduce the heuristic.
2. If not all the digits are used, then with some probability p, we swap out the letter that is most conflicted with one of the unused digits.

There are many possible solutions for Parts (H), (I) and (J), but they need to be coordinated.

[+4] for reasonable transition based on Parts (H) and (I).
[-2] if some transitions are missing, i.e., if $m < 10$ and there is no operation to allow the unused digit to be swapped in.

Question 1L Which is better: uninformed search or local search? [3 marks]

The answer is that it depends. If n is large, then it is likely that there are many constraints that will allow efficient pruning, so uninformed search is better. If n is small and there are many possible solutions, then local search would likely be very fast.

[+1] for it depends.
[+2] for explaining "it depends" correctly.
Most students completely didn't get the "it depends," so instead of marking everyone wrong, we were charitable and gave credit that students who said something relatively sensible.

Question 2A Is h_1 admissible? [2 marks]

Tests that student understands that an admissible heuristic for a relaxed problem must also be admissible for a more constrained problem.
[2m] Yes it is admissible, since this is the exact solution to the related problem where the coin can be directly moved to its goal position.

Question 2B Is h_2 admissible? [2 marks]

This question tests that the student understands if an admissible heuristic must not overestimate the actual distance to the goal.
[2m] Yes it is admissible, since this never overestimates the distance of moving a coin to its goal position. It is a stricter version of the square grid since (1,3) has to take an actual path of 3 time steps to reach (2,3).

Question 2C Maximum Branching Factor [2 marks]

$3k$, since each of the k open spaces can possibly accept a maximum of 3 different tiles.

[2m] state $3k$ with explanation
[1m] correct observations but incorrect final answer, or only state.
[0m] incorrect answers without explanation

Question 2D h_A admissible and consistent? Explain. [4 marks]

○ Admissible ● Not admissible

h_A is not admissible because suppose we have one coin adjacent to an open goal position, and another coin at the other (far) end of the puzzle, also adjacent to an open goal position.
c...c
g...g
Then the heuristic will overestimate the true value 2.

○ Consistent ● Not consistent

h_A is not consistent because it is possible for the heuristic to decrease by more than 1 (the actual cost) if a coin moves to its adjacent open goal position as in the scenario given above.

Question 2E h_B admissible and consistent? Explain. [4 marks]

● Admissible ○ Not admissible

h_B is admissible because the actual distance to the goal state must at least minimally be solved by moving all misplaced coins to their closest goal locations.

○ Consistent ● Not consistent

h_B is not consistent because it is possible for the heuristic to decreases by more than 1 (the actual cost) with each time step, when the coins in C_{open} share the same open goal position, and the open goal position moves closer to the cluster of C_{open}.

Question 2F h_C admissible and consistent? Explain. [4 marks]

○ Admissible ● Not admissible

h_C is inadmissible because this sums the distances of each coin to the most distant goal position; but the optimal solution might require the coin to be placed at a closer goal position instead, causing this heuristic to overestimate.

○ Consistent ● Not consistent

h_C is inconsistent because the heuristic decreases by more than 1 (the actual cost) in a time step, if a coin is placed on a far-away goal position.

Question 2G Dominant relationships between the heuristics. [5 marks]

h_C is clearly dominant over h_B since it takes the maximum distance.
h_C is also dominant over h_A since h_A only takes the minimum of all the maximum distances, while h_C takes the sum of the maximum distances.

[+1] $h_C > h_B$
[+1] $h_C > h_A$
[+1] No dominant relationship between h_A and h_B

Between h_B and h_A, there is no dominant relationship since it is possible for h_B to sometimes be greater than h_A when C is all near the top of the puzzle, and G is all near the bottom, and sometimes h_A to sometimes be greater than h_B when there are only two coins as in the configuration described in 2D's answer.

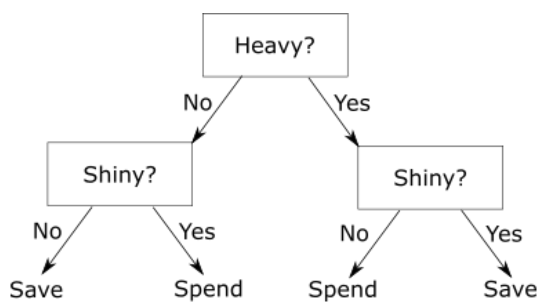
Question 3A To save or to spend? [2 marks]

Ben should save the coin.
If the coin is small, the decision tree shows that Ben should save.
If the coin is big, then it is also heavy and shiny, hence should save too.

This question tests that the student knows how to reason about a decision tree.

Question 3B Minimal-Depth Tree. [2 marks]

This question is here to test that the student knows how to reconfigure a decision tree.



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

The counts of the 10 decisions are 5 for Save and 5 for Spend.
Since there are two classes of equal distribution, the entropy can simply be observed to be 1.

$$\begin{aligned}\text{Entropy} &= -\frac{5}{10}\log(\frac{5}{10}) - \frac{5}{10}\log(\frac{5}{10}) \\ &= +0.5 + 0.5 = 1.00\end{aligned}$$

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

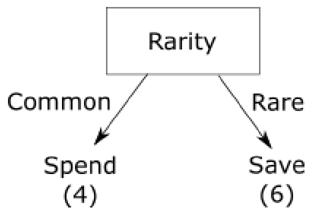
We only need to consider the smallest remaining entropy. In this case, we will use Rarity of the coin to split.

$$\begin{aligned}\text{Remainder(Rarity)} &= \frac{6}{10}I(\frac{1}{6},\frac{5}{6}) + \frac{4}{10}I(\frac{4}{4},0) \\ &= -0.6(\frac{1}{6}\log_2(\frac{1}{6}) + \frac{1}{5}\log_2(\frac{1}{5})) - 0 \\ &= 0.390 \approx 0.39\end{aligned}$$

$$\begin{aligned}\text{Remainder(Vintage)} &= \frac{5}{10}I(\frac{3}{5},\frac{2}{5}) + \frac{5}{10}I(\frac{2}{5},\frac{3}{5}) \\ &= -0.5(\frac{3}{5}\log_2(\frac{3}{5}) + \frac{2}{5}\log_2(\frac{2}{5})) \times 2 \\ &= 0.970 \approx 0.97\end{aligned}$$

$$\begin{aligned}\text{Remainder(Condition)} &= \frac{7}{10}I(\frac{4}{7},\frac{3}{7}) + \frac{3}{10}I(\frac{1}{3},\frac{2}{3}) \\ &= -0.7(\frac{4}{7}\log_2(\frac{4}{7}) + \frac{3}{7}\log_2(\frac{3}{7})) - 0.3(\frac{1}{3}\log_2(\frac{1}{3}) + \frac{2}{3}\log_2(\frac{2}{3})) \\ &= 0.965 \approx 0.97\end{aligned}$$

$$\begin{aligned}\text{Remainder(Production)} &= \frac{2}{10}I(\frac{1}{2},\frac{1}{2}) + \frac{8}{10}I(\frac{4}{8},\frac{4}{8}) \\ &= 1.00\end{aligned}$$



Question 3E Full Decision Tree. [5 marks]

Using the results from part (D), we split on rarity, so we now only consider “Rare” coins.

Coin	Is Vintage	Condition	In Production	Decision
1	No	Rusty	Inactive	Spend
2	No	Rusty	Active	Save
3	Yes	Rusty	Active	Save
4	Yes	Rusty	Active	Save
9	No	Mint	Active	Save
10	Yes	Mint	Inactive	Save

Again, consider the smallest remaining entropy. In this case, we will use Production next.

$$\begin{aligned}\text{Remainder(Vintage)} &= \frac{3}{6}I(\frac{2}{3},\frac{1}{3}) + \frac{3}{6}I(\frac{3}{3},0) \\ &= -0.5(\frac{2}{3}\log_2(\frac{2}{3}) + \frac{1}{3}\log_2(\frac{1}{3})) \\ &= 0.459 \approx 0.46\end{aligned}$$

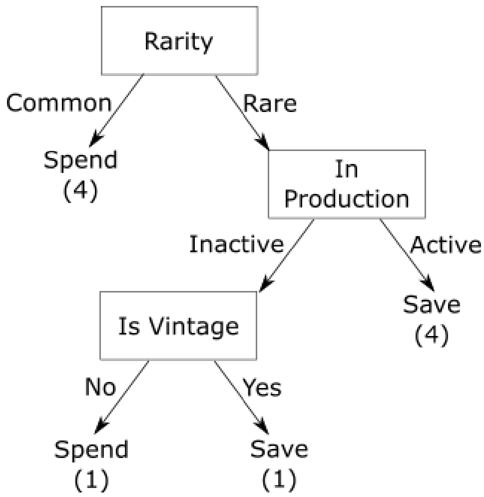
$$\begin{aligned}\text{Remainder(Condition)} &= \frac{4}{6}I(\frac{3}{4},\frac{1}{4}) + \frac{2}{6}I(\frac{2}{2},0) \\ &= -\frac{2}{3}(\frac{3}{4}\log_2(\frac{3}{4}) + \frac{1}{4}\log_2(\frac{1}{4})) \\ &= 0.540 \approx 0.54\end{aligned}$$

$$\begin{aligned}\text{Remainder(Production)} &= \frac{2}{6}I(\frac{1}{2},\frac{1}{2}) + \frac{4}{6}I(\frac{4}{4},0) \\ &= 0.333 \approx 0.33\end{aligned}$$

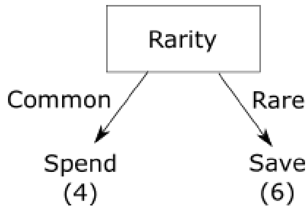
And are now left with only “Rare and Inactive” coins.

Coin	Is Vintage	Condition	Decision
1	No	Rusty	Spend
10	Yes	Mint	Save

Both provide the same information gain, so we tie-break, and use Vintage.



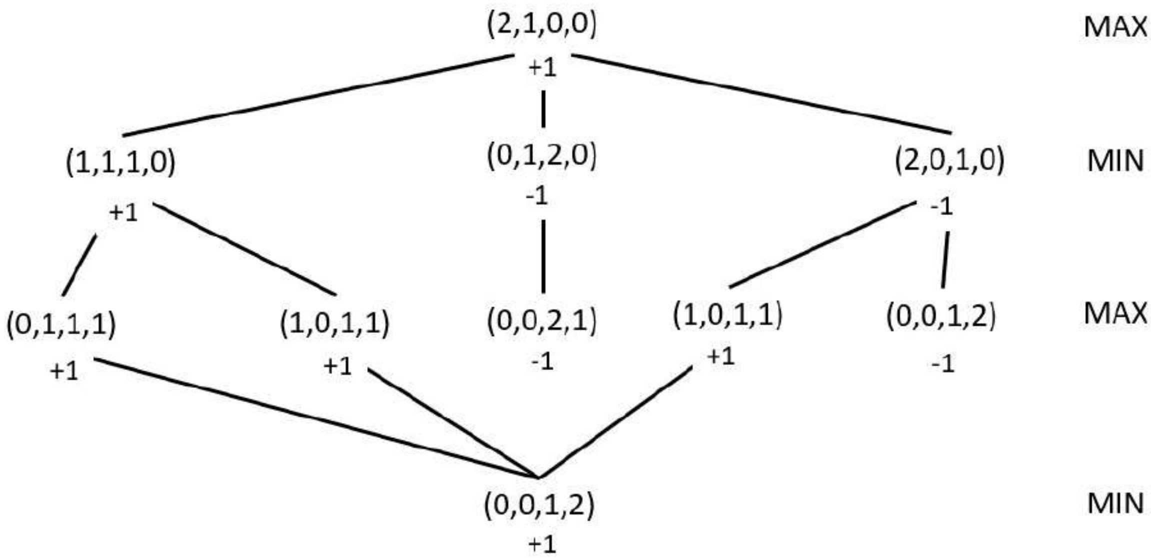
Question 3F Min-Sample DT [4 marks]



Coin 1 is likely an outlier.

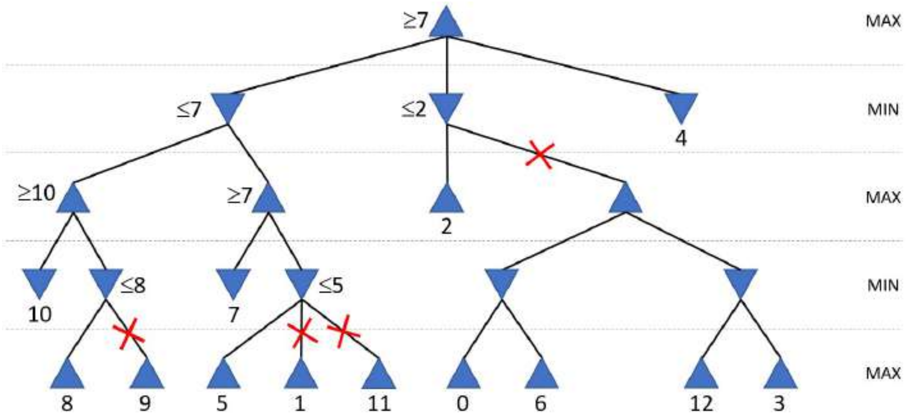
Question 4A Nim Game Tree. [6 marks]

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. First player always wins.



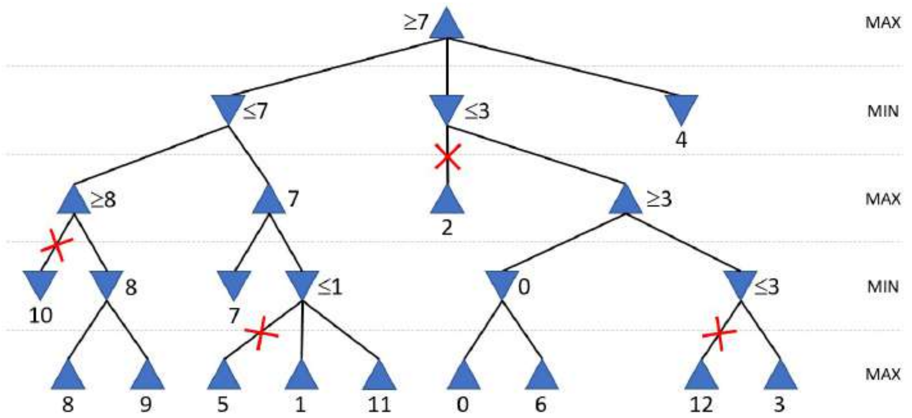
[+3] for the correct tree and state representation.
[+1] for stating that first player wins.
[+2] for solving the game tree correctly.
[-2] for each error.

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



Root value is 7.

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



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.