National University of Singapore

CS2109S—Introduction to AI and Machine Learning

MIDTERM ASSESSMENT

Semester 1, 2024/2025

Time allowed: 1 hour 30 minutes

INSTRUCTIONS TO STUDENTS

1. Please review the exam instructions provided in Examplify.
2. This paper contains the context for the questions in Examplify.
3. This paper consists of **(31) pages** including this cover page.
4. This paper is for your reference only and should not be submitted.
5. All questions must be answered in the Examplify.
6. You may refer to the appendix in Examplify while answering questions.

This page is intentionally left blank.

It may be used as scratch paper.

Question 1: Transwarp Conduits

After years of research into the Borg's transwarp conduits, the Federation is planning to construct its own transwarp conduits. A transwarp conduit can be thought of as a connector or pipe that links a pair of gateways. The coordinates of the two gateways for a single conduit i are denoted by g_i^1 and g_i^2, where $g_i^1 = (x_i^1, y_i^1)$ and $g_i^2 = (x_i^2, y_i^2)$, representing the (x, y) coordinates of the gateways. This project involves establishing N transwarp conduits between pairs of gateways located at coordinates $(g_1^1, g_1^2), \ldots, (g_N^1, g_N^2)$. However, the conduits cannot intersect or branch off, and the gateways must be positioned at the ends of these conduits. The connection of the conduit must either be horizontal or vertical, but not diagonal. Furthermore, the space designated for this construction is highly restricted, with all conduits needing to fit within a tightly controlled $M \times M$ square. The Vulcan Science Directorate has ensured that there is exactly zero or one configuration that fits within the designated space.

An example of a 4x4 transwarp conduits problem instance is illustrated in the figure below. (1) displays the initial state, while (2) presents a valid solution to the problem.

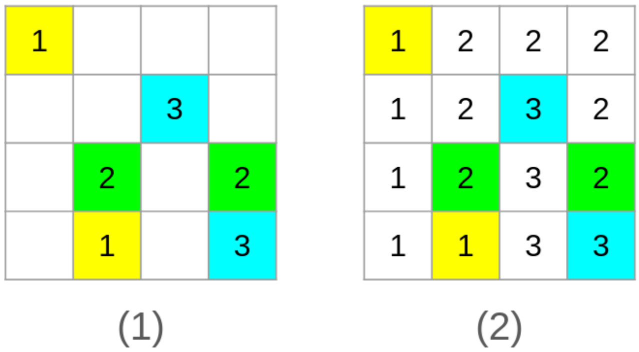


Figure 1: 4x4 Transwarp Conduits Problem

As the head of engineering for the Federation's Transwarp Project, Scotty is responsible for designing the conduit configurations not only for a specific region but also for other regions. Drawing on the knowledge gained from CS2109S course at Starfleet Academy, Scotty has decided to leverage search algorithms to determine the optimal configurations for the conduits.

After many days of sleepless night, Scotty finally settled with the following formulation.

State representation:

- $M \times M$ integer matrix T, where $T_{x,y} \in [0, \ldots, N]$ for any coordinate $(x, y), 1 \leq x \leq M, 1 \leq y \leq M$. Here, $1, \ldots, N$ is the conduit index. $T_{x,y} = 0$ means the coordinate does not have any conduit.

Initial states:

- $\forall_{i,j} T_{x_i^j, y_i^j} = i$, where $(x_i^j, y_i^j) = g_i^j$ is the coordinate of gateway j of the i-th conduit, where $j \in [1, 2]$ and $i \in [1, \ldots, N]$, otherwise 0.

Goal test:

- For each pair of gateways (g_i^1, g_i^2), there is a path $s_1, ..., s_P$ where $s_1 = g_i^1$, $s_P = g_i^2$, $s_j = (x_j, y_j)$, and $T_{x_j, y_j} = i$ for all $j \in [1, ..., P]$

Actions:

- For any arbitrary coordinate (x, y), set $T_{x,y} = c$, where $1 \leq c \leq N$ (N number of actions)

Note: in this question group, a problem formulation is considered reasonable if at least one search algorithm can find the optimal solution if it exists.

Based on the search formulation that Scotty came up with for the **Transwarp Conduits** problem, answer questions 1A-1D.

1A. [3 marks] Based on the search formulation, which of the following invariant(s) is/are (implicitly) satisfied at all times?

a. $T_{x,y} \in [1, ..., N]$ for any coordinate (x, y)

b. For any two points p_i^1 and p_i^2, there is a path $s_1, ..., s_P$ where $s_1 = p_i^1$, $s_P = p_i^2$, $s_j = (x_j, y_j)$, and $\forall_j T_{x_j, y_j} = i$

c. For each pair of gateways (g_i^1, g_i^2), $T_{x_i^1, y_i^1} = T_{x_i^2, y_i^2} = i$

d. None of the above [Correct]

1B. [3 marks] Suppose that you want to change the initial state of the search formulation. Which of the following initial state(s) is/are reasonable?

a. Random initialization, i.e., $T_{(x_i, y_i)} = k$, where k is sampled uniformly from $[1, N]$ [Correct]

b. Random initialization with possible empty coordinates, i.e., $T_{(x_i, y_i)} = k$, where k is sampled uniformly from $[0, N]$ [Correct]

c. Empty initialization, i.e., $T_{(x_i, y_i)} = 0$, for all i [Correct]

d. None of the above

1C. [3 marks] Suppose that you want to change the goal test of the search formulation. Which of the following goal test(s) is/are reasonable?

a. For any two points p_i^1 and p_i^2, there is a path $s_1, ..., s_P$ where $s_1 = p_i^1$, $s_P = p_i^2$, $s_j = (x_j, y_j)$, and $T_{x_j, y_j} = i$ for all $j \in [1, ..., P]$

b. For each pair of gateways (g_i^1, g_i^2), there is a path $s_1, ..., s_P$ where $s_1 = g_i^1$, $s_P = g_i^2$, $s_j = (x_j, y_j)$, and $T_{x_j, y_j} = T_{x_i^1, y_i^1}$ for all $j \in [1, ..., P]$

c. None of the above [Correct]

1D. [4 marks] Suppose that you want to change the actions of the search formulation. Which of the following action(s) is/are reasonable?

a. Select any empty coordinate (x,y), set $T_{x,y} = c$, where $1 \leq c \leq N$ (N number of actions) [Correct]

b. Select any non-empty coordinate $s_i = (x_i, y_i)$, set the value of one of its neighbors n_j to $T_{(x_i,y_i)}$. (Number of actions = number of neighbors) [Correct]

c. Select any non-empty coordinate $s_i = (x_i, y_i)$, set the value of one of its **empty** neighbors n_j to $T_{(x_i,y_i)}$. (Number of actions = number of neighbors) [Correct]

d. Swap $T_{(x_i,y_i)}$ and $T_{(x_j,y_j)}$ for any i, j. ($(M \times M)^2$ number of actions)

e. Perform both (a) and (d) [Correct]

f. Perform both (b) and (d) [Correct]

g. Perform both (c) and (d) [Correct]

h. None of the above

Explanations:

1A

- (a) is incorrect because a cell can have a value of 0, such as during initialization.
- (b) is incorrect since this condition only holds true in the goal state.
- (c) is incorrect because the action may override the value at the gateway's coordinates to something other than i, which represents the index of the gateway.

1B

Since actions can override values at any coordinate, the initialization is irrelevant. Therefore, (a), (b), and (c) are correct.

1C

- (a) is clearly incorrect, as it implies that all values along the path between any two coordinates must be identical, which is only achievable if all coordinates share the same value.
- (b) incorrectly asserts that the values along the path between each pair of gateways must match the value of the first gateway's coordinate. This is false because actions allow for values at any coordinate, including gateways, to be overridden. For instance, consider two pairs of gateways. After executing some actions, the values at the coordinates of both the first and second pairs are changed to one, while the rest of the values are also one. In this scenario, the goal test would erroneously classify this as a goal state. The actual goal state should have the first pair of gateways assigned a value of 1 and the second pair a value of 2, with paths connecting each pair of gateways that share the same values.

1D

- (a) is correct because it incrementally fills the board with numbers. Given that the goal state is reachable from the initial state, a sequence exists that leads to the goal state.
- (b) is correct as it effectively replicates the original action, but incrementally overrides the coordinates surrounding the non-empty coordinates.
- (c) is correct because it similarly fills the coordinates surrounding the non-empty coordinates incrementally.
- (d) is incorrect; swapping coordinates does not result in filling all coordinates.
- (e), (f), and (g) are correct, as they build on the correctness of (a), (b), and (c). Introducing additional options, such as swapping, does not alter this correctness.

Question 2: Transwarp Conduits: Analysis

Analyze the search formulation that Scotty came up with for the **Transwarp Conduits** problem and answer questions 2A-2G.

2A. [1 mark] Is it true that the search formulation results in a state space where a state **can** be **visited multiple times**? Note: we do not care about the search algorithms in this question since we are asking about state space, not search tree/graph

a. Yes [Correct]

b. No

2B. [1 mark] Does the search formulation result in **many goal states**?

a. Yes

b. No [Correct]

2C. [2 marks] Which of the following tree search algorithm(s) can we employ such that the search tree is **finite** (if there is a solution)?

a. Breadth-first search (BFS) [Correct]

b. Depth-first search (DFS)

c. Uniform-cost search (UCS) with cost $c > 1$ for all actions [Correct]

d. Depth-limited search with BFS and max-depth = $M^2 - 2N$ [Correct]

e. Depth-limited search with DFS and max-depth = $M^2 - 2N$ [Correct]

f. Iterative deepening search (IDS) with DFS [Correct]

g. None of the above

2D. [2 marks] Which of the following tree search algorithm(s) can we employ such that the search **always terminates** (including if there is no solution)?

a. Breadth-first search (BFS)

b. Depth-first search (DFS)

c. Uniform-cost search (UCS) with cost $c > 1$ for all actions

d. Depth-limited search with BFS and max-depth = $M^2 - 2N$ [Correct]

e. Depth-limited search with DFS and max-depth = $M^2 - 2N$ [Correct]

f. Iterative deepening search (IDS) with DFS

g. None of the above

2E. [2 marks] Which of the following tree search algorithm(s) can we employ such that the search **always finds an answer** (valid solution) if a solution exists?

a. Breadth-first search (BFS) [Correct]

b. Depth-first search (DFS)

c. Uniform-cost search (UCS) with cost $c > 1$ for all actions [Correct]

d. Depth-limited search with BFS and max-depth = $M^2 - 2N$ [Correct]

e. Depth-limited search with DFS and max-depth = $M^2 - 2N$ [Correct]

f. Iterative deepening search (IDS) with DFS [Correct]

g. None of the above

2F. [2 marks] Which of the following **tree search** algorithm(s) is/are the **best** for the problem?

Best means the algorithm(s) should be complete, optimal, efficient (in terms of big O worstcase space and time complexity), and aware if there is no solution.

a. Breadth-first search (BFS)

b. Depth-first search (DFS)

c. Uniform-cost search (UCS) with cost $c > 1$ for all actions

d. Depth-limited search with BFS and max-depth = $M^2 - 2N$

e. Depth-limited search with DFS and max-depth = $M^2 - 2N$ [Correct]

f. Iterative deepening search (IDS) with DFS

g. None of the above

2G. [2 marks] Suppose that we use graph search instead of tree search. Which of the following **graph search** algorithm(s) is/are the **best** for the problem?

Best means the algorithm(s) should be complete, optimal, efficient (in terms of big O worstcase space and time complexity), and aware if there is no solution.

a. Breadth-first search (BFS)

b. Depth-first search (DFS) [Correct]

c. Uniform-cost search (UCS) with cost $c > 1$ for all actions

d. Depth-limited search with BFS and max-depth = $M^2 - 2N$

e. Depth-limited search with DFS and max-depth = $M^2 - 2N$ [Correct]

f. Iterative deepening search (IDS) with DFS [Correct]

g. None of the above

Explanations:

2A

Actions can override any coordinates, allowing the same coordinate to be modified repeatedly. This can result in revisiting the same state multiple times.

2B

As stated in the question, "there is exactly zero or one configuration that fits ...". Therefore, if a goal state exists, there can only be one.

2C

- BFS, DFS, and IDS will find the solution and terminate since they search depth-by-depth, and the solution is located at a finite depth. Consequently, the search tree is finite.
- DFS may not terminate due to the reversibility of actions, leading to an infinite search tree.
- DLS will terminate once it reaches the maximum depth, making the search tree finite.

2D

Only DLS guarantees termination. If no solution exists, other algorithms may not recognize when to stop and could continue searching indefinitely.

2E

Refer to the explanation in 2C. For DLS, we set the maximum depth to the depth of the solution, which corresponds to the depth at which all cells are filled. Given M^2 total cells and $2N$ coordinates occupied by the N pairs of gateways in the initial state, there are $M^2 - 2N$ cells left to fill.

2F

- BFS, UCS, and DLS with BFS are not space efficient, requiring $O(b^d)$ space.
- DFS may not terminate.
- DLS and IDS share the same **worst-case** space and time complexity.
- BFS, UCS, and IDS will not terminate if there is no solution.

Thus, DLS is the best.

2G

Refer to explanation 2F. However, in this context, DFS will always terminate.

Alternative Interpretation:

There are two acceptable interpretations of the actions:

- Default Interpretation: For any arbitrary coordinate, set its value. There are M^2 possible coordinates and N possible values. Thus, there are N possible values (actions) per coordinate which results in $M^2 \times N$ total actions.
- Alternative Interpretation: For any arbitrary (randomly selected) coordinate, set its value. Since there is only one coordinate selected and there are N possible values, the total number of actions is N.

The answers and explanations provided earlier follow the default interpretation. Here, we will present the answers and explanations based on the alternative interpretation.

2C

Since coordinates are chosen randomly, it's possible to select the same coordinate multiple times. As a result, the search tree for any tree search algorithm without a depth limit could potentially become infinite. This is because the coordinate could be selected repeatedly, leading to infinite search tree.

Only Depth-Limited Search (DLS) would correctly handle this scenario.

2E and 2F

The answer is "None of the above". Please refer to the explanation for 2C for further clarification on why this is the case.

Question 3: Minimum-Region Transwarp Conduits

Pleased with the results of the initial transwarp conduits project, the Federation has tasked Scotty with constructing more conduits. This time, there are no restrictions on the size of the region; they simply want the conduits to utilize space as efficiently as possible.

The federation defines space utilization $U = |a-b| \times |c-d|$, where a and b are the farthest non-empty points along the x-axis, and c and d are the farthest non-empty points along the y-axis.

For this task, Scotty plans to employ a local search strategy, which he formulates as follows:

State Representation:

- An $M \times M$ integer matrix T, where $T_{x,y} \in [0,...,N]$ for any coordinate (x,y) such that $1 \leq x \leq M$ and $1 \leq y \leq M$. A value of 0 indicates that the coordinate is empty.

Initial State:

- For each pair of gateways g_i^1 and g_i^2, set $T_{x_j,y_j} = i$ for each (x_j, y_j) on a randomly chosen non-intersecting path between g_i^1 and g_i^2.

Successor Function:

- Select a pair of gateways g_i^1 and g_i^2, select $p > 0$ random non-overlapping subpaths in the path between g_i^1 and g_i^2, and replace each one with a new random non-intersecting subpath. If g_i^1 and g_i^2 are not connected, then create a random non-intersecting path between the two.

Evaluation Function:

- Negative space utilization U + total number of neighboring coordinates of each gateway which have the same value as the gateway

Note:

- In this context, we consider the formulation reasonable if hill-climbing with an infinitely large number of random restarts can reach the global optimum.
- If there are no non-empty points, then Federation space utilization measure return 0
- Non-intersecting (sub)path means that the path is not intersecting other conduits

Based on the description, answer questions 3A-3D.

3A. [3 marks] Suppose that you want to change the initial state of the search formulation. Which of the following initial state(s) is/are reasonable?

a. For each pair of gateways g_i^1 and g_i^2, set $T_{x_j,y_j} = i$ for each (x_j, y_j) on a randomly chosen path between g_i^1 and g_i^2.

b. Random initialization, i.e., $T_{(x_i,y_i)} = k$, where k is sampled uniformly from $[1,N]$

c. Random initialization with possible empty coordinates, i.e., $T_{(x_i,y_i)} = k$, where k is sampled uniformly from $[0,N]$

d. Empty initialization, i.e., $T_{(x_i,y_i)} = 0$, for all i

e. None of the above [Correct]

3B. [4 marks] Suppose that you want to change the successor function of the search formulation. Which of the following successor function(s) is/are reasonable?

a. Select a pair of gateways g_i^1 and g_i^2, select up to $p > 0$ random non-overlapping subpaths along the connecting path and replace each one with a random non-intersecting subpath.

b. Select any non-empty coordinate $s_i = (x_i, y_i)$, set the value of one of its neighbors n_j to $T_{(x_i,y_i)}$. (Number of successors = number of neighbors)

c. Select any non-empty coordinate $s_i = (x_i, y_i)$, set the value of one of its **empty** neighbors n_j to $T_{(x_i,y_i)}$. (Number of successors = number of neighbors)

d. Swap $T_{(x_i,y_i)}$ and $T_{(x_j,y_j)}$ for any i, j. ($M \times M$ number of successors)

e. Perform both (b) and (d)

f. Perform both (c) and (d)

g. None of the above [Correct]

3C. [3 marks] Suppose that you want to change the evaluation function of the search formulation. Which of the following evaluation function(s) should you add (sum) together so that the formulation is reasonable?

a. Number of non-empty cells

b. Max length of conduits

c. Average Length of conduits

d. Space utilization U

e. Number of intersecting paths between gateways

f. Number of connected gateways

g. No combinations will lead to a reasonable formulation [Correct]

3D. [3 marks] Suppose that we use the search formulation that Scotty came up with for the **Transwarp Conduits** problem.

Which of the following uninformed search algorithm(s) can we use to find the optimal solution for the **Minimum-Region Transwarp Conduits** problem assuming that it exists?

a. Breadth-first search (BFS)

b. Depth-first search (DFS)

c. Uniform-cost search (UCS) with constant cost $c > N$

d. Depth-limited search with BFS and max-depth = $(N-1)^2 - 2N$

e. Depth-limited search with DFS and max-depth = $(N-1)^2 - 2N$

f. Iterative deepening search (IDS) with DFS

g. None of the above [Correct]

Explanations:

Here, we make two somewhat obvious assumptions. First, we assume that M is sufficiently large. Second, we assume that the initial state is not the goal state; if it were, there would be no need for local search.

3A and 3B

The problem formulation states that we initialize the state by connecting each pair of gateways through random, non-intersecting paths. The successor functions then generate new, random non-intersecting paths. However, the evaluation function is somewhat problematic. The second term, which counts the total number of neighboring coordinates of each gateway which have the same value as the gateway, contradicts the overall objective. This can lead to a suboptimal state being incorrectly deemed optimal by the evaluation function.

For example, consider a pair of gateways located at (0,0) and (0,3). If we fill in the coordinates (0,0), (0,1), (0,2), (0,3), (1,0), (1,1), (1,2), and (1,3), the evaluation function yields a value of -8 + 6 = -2. This value matches that of the optimal solution, which includes the coordinates (0,0), (0,1), (0,2), and (0,3): -4 + 2 = -2.

We define the formulation as reasonable if hill-climbing with an infinite number of random restarts can reach the global optimum (and therefore return the optimal solution). However, in this case, the solution returned by hill-climbing with infinite random restarts may be incorrect because the evaluation function cannot differentiate between suboptimal and optimal solutions.

Any modifications to the initial state (3A) and the successor function (3B) will not resolve this issue.

3C

- (a), (b), and (c) are incorrect as they fail to assess the efficiency of space utilization.
- (d) should not be maximized.
- (e) is unnecessary since the successor function guarantees that there are no intersecting

paths.

- (f) is a constant because both the initial state and the successor function ensure that all gateways are consistently connected, making it redundant.

3D

In this modified problem, fewer steps (search depth) do not necessarily indicate a better solution. Consider an instance with the pairs of gateways ((0,0),(2,0)), ((1,2),(1,4)), and ((0,5),(1,5)). The optimal solution involves setting the coordinates (0,0), (1,0), (2,0) to 1, the coordinates (1,2), (1,3), (1,4) to 2, and the coordinates (0,5), (0,4), (0,3), (0,2), (0,1), (1,1), (2,1), (2,2), (2,3), (2,4), (2,5), (1,5) to 3, resulting in a total of 12 steps (since the gateways are filled in the initial state).

In contrast, another solution connects the third gateway by setting (0,5), (0,6), (1,6), (1,5) to 1. While this approach requires only 4 steps, it leads to greater space utilization, covering 21 cells compared to the 18 cells utilized in the optimal solution. Thus, BFS and IDS would fail to return the optimal solution. UCS, using a constant cost c, operates similarly to BFS. For DLS, the maximum depth is set incorrectly. As a result, none of the algorithms will find the optimal solution.

Question 4: Fruit

There are $n \times n$ rooms on a farm, with fruit located in different rooms. Initially, the number of rooms containing fruit is more than one. Your task is to gather all the fruit into the room located at (0, 0) and move yourself to that room as well. To move the fruit, you need to enter the room and move together with it. The position of each room, r, is represented by (x_r, y_r). x_r and y_r represent the row and column index of room r. You are denoted by p, with your current position being (x_p, y_p). You cannot move outside the $n \times n$ grid.

In each step, you can make the following moves:

1. Move yourself one room left, right, up, or down at a cost of **1**.
2. Move yourself along with **all** the fruit in a room one step left, right, up, or down at a cost of **2**.

For instance, consider Figure 2 (1), where the positions of the rooms with fruit are {(1, 1), (4, 3)}, and your current position is (1, 1). You can choose to move yourself to one of the adjacent rooms at a cost of 1, as shown in Figure 2 (2). Additionally, you can opt to move along with all the fruit in room located at (1, 1) at a cost of 2, as illustrated in Figure 2 (3). Your goal, as shown in Figure 2 (4), is to move both yourself and all the fruit to the room located at (0, 0).

After discussing this problem with your friend, he/she proposed two heuristics to be used with the A* search algorithm.

$\boldsymbol{h_A}$: Twice the number of rooms with fruit that are not at position (0, 0).

$$h_A = n_f * 2 \tag{1}$$

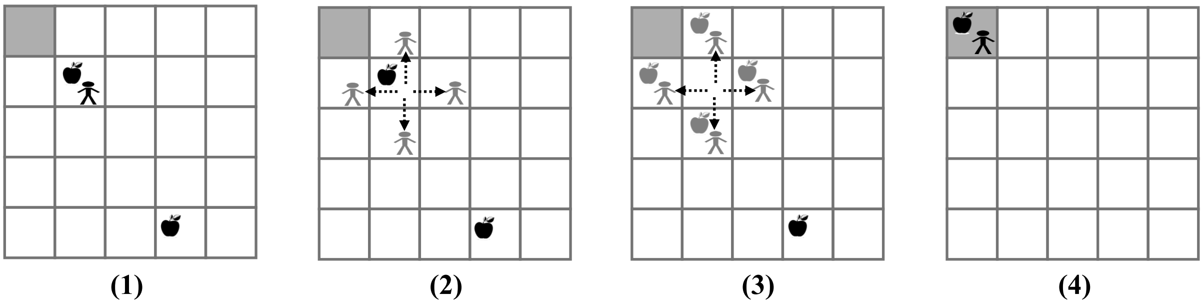


Figure 2: Fruit

Here, n_f denotes the number of rooms with fruit that are not at position (0, 0). For example, if there are 5 rooms with fruit and the target room (0, 0) is empty, then $n_f = 5$. However, if the target room contains fruit, then $n_f = 5 - 1 = 4$.

h_B: The sum of the Manhattan distances between the rooms with fruit and your current position.

$$h_B = \sum_{r \in F} MD(r, p) \tag{2}$$

Here, F contains all rooms with fruit. You are denoted by p.

Hint: The Manhattan Distance is defined as $MD(a,b) = |x_a - x_b| + |y_a - y_b|$, where x_a and y_a represent the row and column index of a, and the same applies to b.

Based on the description, answer questions 4A-4B.

4A. [3 marks] Is h_A admissible for the Fruit problem?

a. Yes, it is admissible. [Correct]

b. No, it is not admissible.

4B. [3 marks] Is h_B admissible for the Fruit problem?

a. Yes, it is admissible.

b. No, it is not admissible. [Correct]

Explanation:
4A. In order to reduce value for n_f by one, the real cost is at least two.

The goal is to make n_f to become 0, so the real cost is at least $n_f * 2$ and h_A is admissible.

4B. Consider a situation where your current position is (3, 3), and positions of rooms with fruit are $\{(1, 0), (2, 0)\}$. $h_B(n) = 5 + 4 = 9$, which exceeds the real cost, $h^* = 4 + 2 + 2 = 8$. Thus, h_B is not admissible.

Question 5: Apple and Banana

Let's consider a modified version of the Fruit game where different types of fruit need to be gathered in separate rooms. Assume there are two types of fruit, apples and bananas, scattered across different rooms. Initially, some rooms may contain both apples and bananas. Your task is to move all the apples to the room G_a located at (0, 0) and all the bananas to the room G_b located at (0, $n-1$). Additionally, you also need to move yourself to the room G_a.

In each step, you can perform the following moves:

1. Move yourself one room left, right, up, or down at a cost of **1**.
2. Move yourself along with **all the bananas / all the apples / all the fruit** in a room one step left, right, up, or down at a cost of **1**.

As shown in Figure 3 (1), the positions of the rooms containing apples are {(1, 1), (3, 2)}, while the positions of the rooms with bananas are {(1, 1), (2, 3)}. Your current position is (1, 1). You can choose to move yourself to one of the four rooms, as depicted in Figure 3 (2). Additionally, you have the option to move along with all the bananas (Figure 3 (3)), all the apples (Figure 3 (4)), or all the fruit (both apples and bananas) (Figure 3 (5)) in the room located at (1, 1). Your goal, as illustrated in Figure 3 (6), is to move both yourself and all the apples to the room at (0, 0), and to place all the bananas in the room at (0, $n-1$).

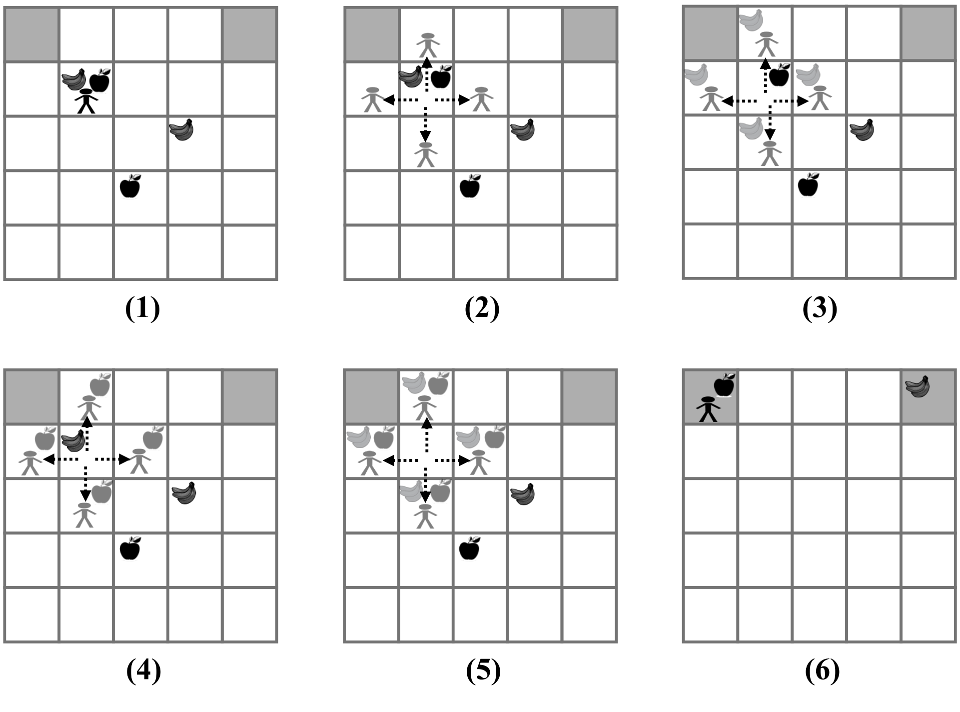


Figure 3: Apple and Banana

After discussing this problem with your friend again, he/she proposed two new heuristics to be used with the A* search algorithm.

$\boldsymbol{h_C}$: The sum of the number of rooms with apple that are not at G_a and the number of rooms with banana that are not at G_b.

$$h_C = n_a + n_b \tag{3}$$

Here, n_a denotes the number of rooms with apple that are not at G_a. n_b denotes the number of rooms with banana that are not at G_b.

$\boldsymbol{h_D}$: The sum of the Manhattan distances between room r_a and G_a, and between room r_b and G_b. Room r_a is the nearest room with apples to G_a, while room r_b is the nearest room with bananas to G_b.

$$h_D = MD(r_a, G_a) + MD(r_b, G_b) \tag{4}$$

$$\text{where} \quad r_a = \arg\min_{r \in S_A} MD(r, G_a) \tag{5}$$

$$r_b = \arg\min_{r \in S_B} MD(r, G_b) \tag{6}$$

Here, S_A contains all rooms with apples and S_B contains all rooms with bananas.

Based on the description, answer questions 5A-5D.

5A. [3 marks] Is h_C admissible for the Apple and Banana problem?

a. Yes, it is admissible.

b. No, it is not admissible. [Correct]

5B. [3 marks] Is h_C consistent for the Apple and Banana problem?

a. Yes, it is consistent.

b. No, it is not consistent. [Correct]

5C. [3 marks] Is h_D admissible for the Apple and Banana problem?

a. Yes, it is admissible.

b. No, it is not admissible. [Correct]

5D. [3 marks] Is h_D consistent for the Apple and Banana problem?

a. Yes, it is consistent.

b. No, it is not consistent. [Correct]

Explanation:

5A. Consider a scenario where rooms with apple are: $\{(0,1),(0,2),(0,3),(1,3)\}$, and rooms with banana are: $\{(1,3)\}$. Your current position is (1, 3). The positions of G_a and G_b are (0, 0) and (0, 3), respectively. In this case, $h_C = 4+1 = 5$. $h^* = 4 < h_C$. Thus, h_C is not admissible.

5B. Consider a scenario where rooms with apple are: $\{(2,3),(3,3)\}$, and rooms with banana are: $\{(2,3),(3,3)\}$. Your current position is (3, 3). The positions of G_a and G_b are (0, 0) and (0, 3), respectively. For the current node n, $h_C(n) = 2+2 = 4$. By moving all the fruit in room at (3, 3) up for one room to (2, 3) at a cost of 1, a successor n' of n can be generated. $h_C(n') = 1 + 1 = 2$. $h_C(n) - h_C(n') = 4 - 2 > 1 = c(n, a, n')$. Thus, h_C is not consistent.

5C. Consider a scenario where rooms with apple are: $\{(2,3),(3,3)\}$, and rooms with banana are: $\{(3,2),(3,3)\}$. Your current position is (3, 2). The positions of G_a and G_b are (0, 0) and (0, 3), respectively. Both G_a and G_b are empty. In this case, $h_D = 5 + 3 = 8$. $h^* = 7 < h_D$. Thus, h_D is not admissible.

5D. Consider a scenario where rooms with apple are: $\{(1,1)\}$, and rooms with banana are: $\{(1,1)\}$. Your current position is (1, 1). The positions of G_a and G_b are (0, 0) and (0, 3), respectively. Both G_a and G_b are empty. For the current node n, $h_D(n) = 2 + 3 = 5$. By moving all the fruit in room (1, 1) up for one room to (0, 1) at a cost of 1, a successor n' of n can be generated. $h_D(n') = 1 + 2 = 3$. $h_D(n) - h_D(n') = 5 - 3 > 1 = c(n, a, n')$. Thus, h_D is not consistent.

Question 6: Predicting Promotion Decision

As an HR professional in a tech company, you have been assigned the task of predicting staff promotion decisions. Drawing on your expertise, you have decided to start the project by constructing a decision tree using the historical data presented in Table 1.

	Experience	Feedback	Potential	Impact	Promotion Decision
0	1-3 years	bad	high	big	yes
1	1-3 years	good	high	big	yes
2	1-3 years	good	low	small	yes
3	1-3 years	good	mid	big	yes
4	< 1 year	bad	low	small	yes
5	< 1 year	bad	mid	big	no
6	< 1 year	good	high	big	no
7	< 1 year	good	high	big	yes
8	< 1 year	good	mid	small	yes
9	> 3 years	bad	high	big	yes

Table 1: Promotion Decision Data

Suppose that you pick "Impact" as the root of your decision tree. Using the data in the table and information gain to split the data, create the full decision tree. In case of a tie, the priority order for constructing the tree is Experience (most preferred) > Feedback > Impact > Potential (least preferred).

Hint: The information content for a given probability distribution p_i, for $i = 1, ..., n$ is given by: Entropy $= -\sum_{i=1}^{n} p_i \log_2(p_i)$.

$\log_2(1) = 0$; $\log_2(2) = 1$; $\log_2(3) = 1.585$; $\log_2(4) = 2$; $\log_2(5) = 2.322$;

$\log_2(6) = 2.585$; $\log_2(7) = 2.807$; $\log_2(8) = 3$; $\log_2(9) = 3.170$; $\log_2(10) = 3.322$

Based on the description, answer questions 6A-6F.

6A. [2 marks] What is the entropy of the Promotion Decision (yes/no) in the table, rounded to 2 decimal places? Hint: remember the log is in base 2! [0.72]

6B. [2 marks] What is root node for the "Impact" = "big" sub-tree?

a. Experience [Correct]

b. Feedback

c. Impact

d. Potential

e. no

f. yes

g. no/yes

6C. [1 mark] According to your full decision tree, what is the Promotion Decision for the following staff?

Staff Information: Experience: < 1 year / Feedback: bad / Potential: high / Impact: big /

a. no [Correct]

b. yes

c. no/yes

6D. [1 mark] According to your full decision tree, what is the Promotion Decision for the following staff?

Staff Information: Experience: > 3 years / Feedback: good / Potential: high / Impact: big /

a. no

b. yes [Correct]

c. no/yes

6E. [1 mark] Please prune the full decision tree ensuring that each leaf node contains at least 2 training data points. How many leaf nodes in your pruned decision tree? [2]

6F. [1 mark] Please prune the full decision tree ensuring that each leaf node contains at least 2 training data points. According to your pruned decision tree, what is the Promotion Decision for the following staff?

Staff Information: Experience: 1-3 years / Feedback: good / Potential: low / Impact: big /

a. no

b. yes [Correct]

c. no/yes

One of your colleagues also designs his/her own decision tree to help with prediction as shown in Figure 4. Your task is to evalute the performance of this decision tree using the data provided in Table 1. Please use "yes" as a positive label and "no" as a negative label.

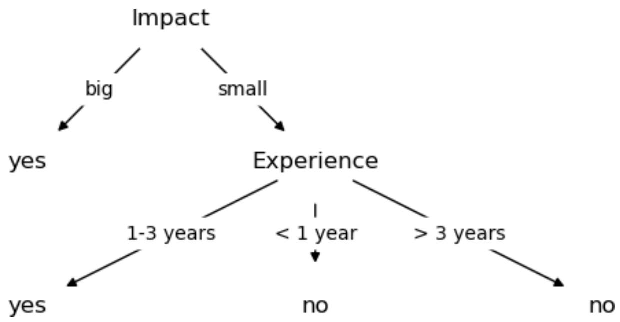


Figure 4: Experts' Decision Tree

Based on the description, answer questions 6G-6J.

6G. [1 mark] What is the number of True Positives (TP)? [6]

6H. [1 mark] What is the number of False Positives (FP)? [2]

6I. [1 mark] What is the number of True Negatives(TN)? [0]

6J. [1 mark] What is the number of False Negatives (FN)? [2]

Explanation:

6A.

$$\begin{aligned} Entropy &= I(\frac{8}{10}, \frac{2}{10}) \\ &= -\frac{8}{10}\log_2(\frac{8}{10}) - \frac{2}{10}\log_2(\frac{2}{10}) \\ &= 0.72 \end{aligned}$$

6B.

If Impact = "big":

	Experience	Feedback	Potential	Promotion Decision
0	1-3 years	bad	high	yes
1	1-3 years	good	high	yes
3	1-3 years	good	mid	yes
5	< 1 year	bad	mid	no
6	< 1 year	good	high	no
7	< 1 year	good	high	yes
9	> 3 years	bad	high	yes

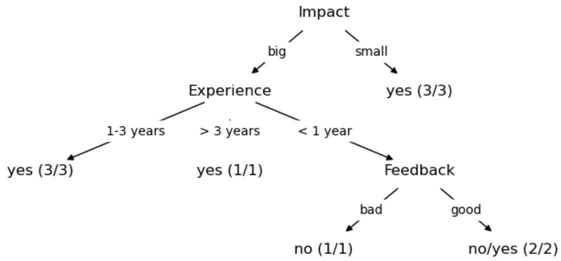
$$
\begin{aligned}
Remainder(Experience) &= \frac{3}{7}\big(I(\frac{3}{3})\big) + \frac{3}{7}\big(I(\frac{2}{3},\frac{1}{3})\big) + \frac{1}{7}\big(I(\frac{1}{1})\big) \\
&= \frac{3}{7}(0.000) + \frac{3}{7}(0.918) + \frac{1}{7}(0.000) \\
&= 0.000 + 0.394 + 0.000 \\
&= 0.394 \\
Remainder(Potential) &= \frac{5}{7}\big(I(\frac{4}{5},\frac{1}{5})\big) + \frac{2}{7}\big(I(\frac{1}{2},\frac{1}{2})\big) \\
&= \frac{5}{7}(0.722) + \frac{2}{7}(1.000) \\
&= 0.516 + 0.286 \\
&= 0.801 \\
Remainder(Feedback) &= \frac{3}{7}\big(I(\frac{2}{3},\frac{1}{3})\big) + \frac{4}{7}\big(I(\frac{3}{4},\frac{1}{4})\big) \\
&= \frac{3}{7}(0.918) + \frac{4}{7}(0.811) \\
&= 0.394 + 0.464 \\
&= 0.857
\end{aligned}
$$

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

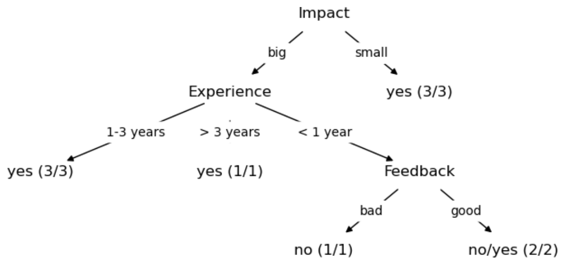
If Experience = "< 1 year", then we split again:

	Feedback	Potential	Promotion Decision
5	bad	mid	no
6	good	high	no
7	good	high	yes

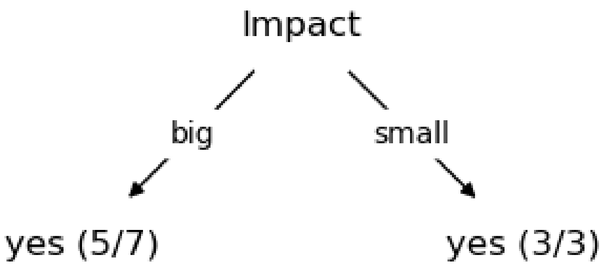
6C.
Promotion Decision: no



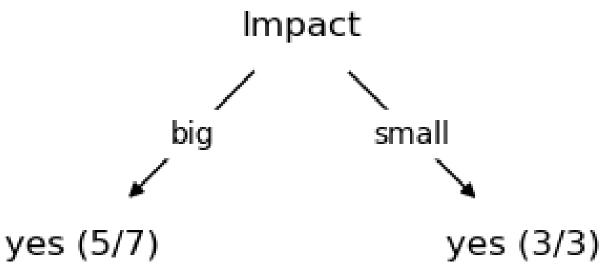
6D.
Promotion Decision: yes



6E.
2 leaf node



6F.
Promotion Decision: yes



Question 7: Gold Coin

Consider a two-player game played on a **1D grid with 4 cells**. The grid contains **three coins**: **two normal coins** located in cell **1** (leftmost) and cell **2**, and one **gold coin** in cell **4** (rightmost). Each player, in turn, can either move any one coin any number of empty spaces to the left or take the coin in the leftmost cell. Players cannot stack coins on the same cell or jump coins past each other. The **winner** is the player who takes the gold coin.

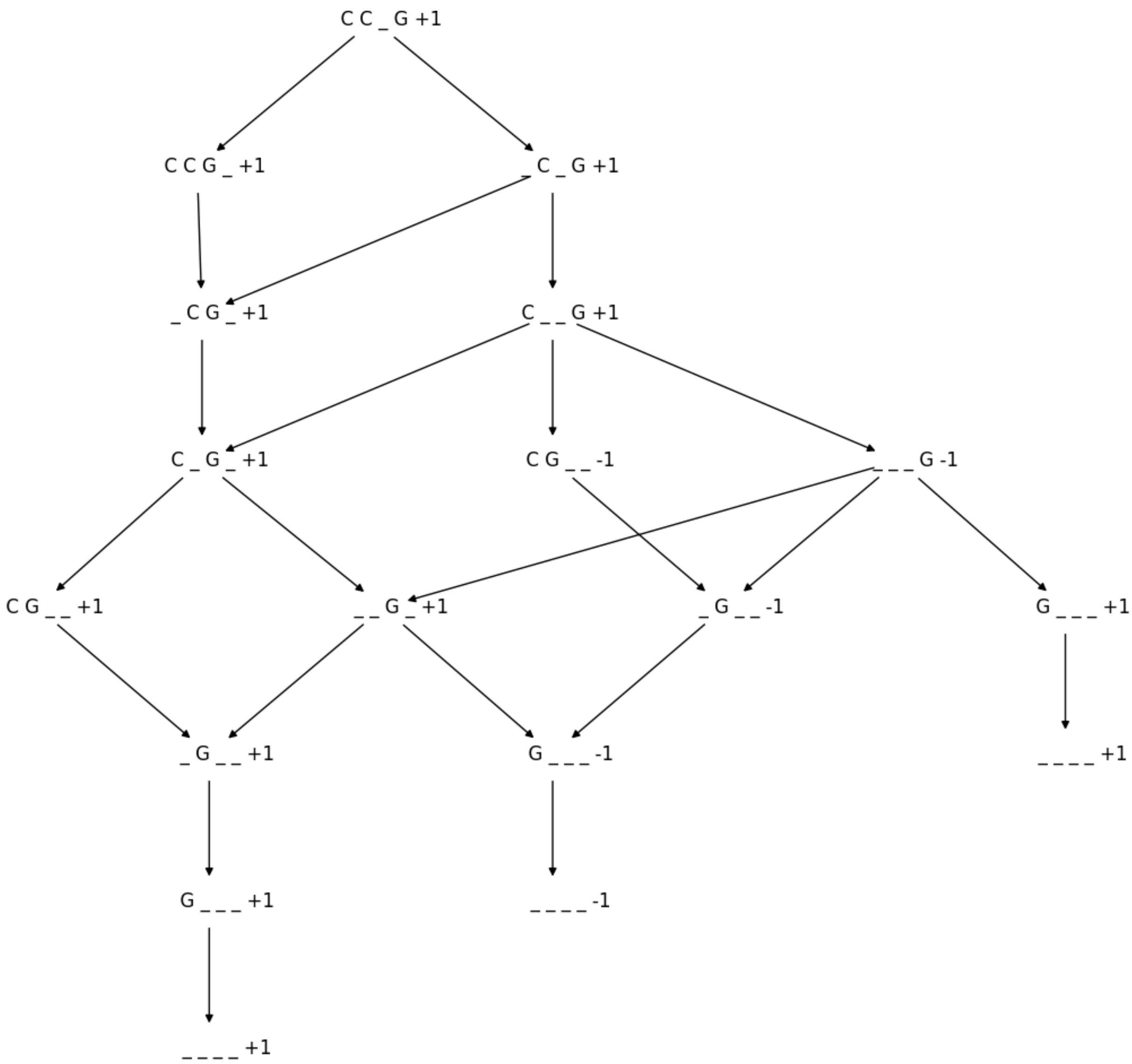
At the start, the first player can choose to either take the coin in the leftmost cell or move the gold coin one step to the left. If the first player moves the gold coin, the second player can only take the coin in the leftmost cell on their turn. The game continues until the gold coin is taken, with the player who takes it declared the winner.

We represent the game state with a **tuple of length 4**, where each element can be "C" (normal coin), "G" (gold coin), or _ (empty cell). Using this representation, the initial state of the game is: "C C _ G". We will use the Minimax algorithm to solve the game by constructing the complete game tree.

Notes:

- We assume that the first player is the max player.
- The value of a state is evaluated from the perspective of the max player.
- In terminal states, a win is valued at +1, a loss at -1, and a draw at 0.

Construct the game tree and answer questions 7A-7Q.



7A. [0.5 mark] Which of the following state(s) is/are the **direct** children of state _ _ _ G? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G.

c. _ _ G _. [Correct]

d. _ C _ G.

e. _ C G _.

f. _ G _ _. [Correct]

g. C _ _ G.

h. C _ G _.

i. C C _ G.

j. C C G _.

k. C G _ _.

l. G _ _ _. [Correct]

m. None of the above

7B. [0.5 mark] Which of the following state(s) is/are the **direct** children of state _ _ G _? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G.

c. _ _ G _.

d. _ C _ G.

e. _ C G _.

f. _ G _ _. [Correct]

g. C _ _ G.

h. C _ G _.

i. C C _ G.

j. C C G _.

k. C G _ _.

l. G _ _ _. [Correct]

m. None of the above

7C. [0.5 mark] Which of the following state(s) is/are the **direct** children of state _ C _ G? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G.

c. _ _ G _.

d. _ C _ G.

e. _ C G _. [Correct]

f. _ G _ _.

g. C _ _ G. [Correct]

h. C _ G _.

i. C C _ G.

j. C C G _.

k. C G _ _.

l. G _ _ _.

m. None of the above

7D. [0.5 mark] Which of the following state(s) is/are the **direct** children of state _ C G _? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G.

c. _ _ G _.

d. _ C _ G.

e. _ C G _.

f. _ G _ _.

g. C _ _ G.

h. C _ G _. [Correct]

i. C C _ G.

j. C C G _.

k. C G _ _.

l. G _ _ _.

m. None of the above

7E. [0.5 mark] Which of the following state(s) is/are the **direct** children of state _ G _ _? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G.

c. _ _ G _.

d. _ C _ G.

e. _ C G _.

f. _ G _ _.

g. C _ _ G.

h. C _ G _.

i. C C _ G.

j. C C G _.

k. C G _ _.

l. G _ _ _. [Correct]

m. None of the above

7F. [0.5 mark] Which of the following state(s) is/are the **direct** children of state C _ _ G? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G. [Correct]

c. _ _ G _.

d. _ C _ G.

e. _ C G _.

f. _ G _ _.

g. C _ _ G.

h. C _ G _. [Correct]

i. C C _ G.

j. C C G _.

k. C G _ _. [Correct]

l. G _ _ _.

m. None of the above

7G. [0.5 mark] Which of the following state(s) is/are the **direct** children of state C _ G _? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G.

c. _ _ G _. [Correct]

d. _ C _ G.

e. _ C G _.

f. _ G _ _.

g. C _ _ G.

h. C _ G _.

i. C C _ G.

j. C C G _.

k. C G _ _. [Correct]

l. G _ _ _.

m. None of the above

7H. [0.5 mark] Which of the following state(s) is/are the **direct** children of state C C _ G? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G.

c. _ _ G _.

d. _ C _ G. [Correct]

e. _ C G _.

f. _ G _ _.

g. C _ _ G.

h. C _ G _.

i. C C _ G.

j. C C G _. [Correct]

k. C G _ _.

l. G _ _ _.

m. None of the above

7I. [0.5 mark] Which of the following state(s) is/are the **direct** children of state C C G _? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G.

c. _ _ G _.

d. _ C _ G.

e. _ C G _. [Correct]

f. _ G _ _.

g. C _ _ G.

h. C _ G _.

i. C C _ G.

j. C C G _.

k. C G _ _.

l. G _ _ _.

m. None of the above

7J. [0.5 mark] Which of the following state(s) is/are the **direct** children of state C G _ _? Select all that is/are true.

a. _ _ _ _.

b. _ _ _ G.

c. _ _ G _.

d. _ C _ G.

e. _ C G _.

f. _ G _ _. [Correct]

g. C _ _ G.

h. C _ G _.

i. C C _ G.

j. C C G _.

k. C G _ _.

l. G _ _ _.

m. None of the above

7K. [0.5 mark] Which of the following state(s) is/are the **direct** children of state G _ _ _? Select all that is/are true.

a. _ _ _ _. [Correct]

b. _ _ _ G.

c. _ _ G _.

d. _ C _ G.

e. _ C G _.

f. _ G _ _.

g. C _ _ G.

h. C _ G _.

i. C C _ G.

j. C C G _.

k. C G _ _.

l. G _ _ _.

m. None of the above

7L. [0.5 mark] What is the value of state _ _ _ _ at depth 6? Note: the root node is at depth 0

a. -1 [Correct]

b. 0

c. 1

d. None of the above

7M. [0.5 mark] What is the value of state C G _ _ at depth 3? Note: the root node is at depth 0

a. -1 [Correct]

b. 0

c. 1

d. None of the above

7N. [0.5 mark] What is the value of state C _ _ G at depth 2? Note: the root node is at depth 0

a. -1

b. 0

c. 1 [Correct]

d. None of the above

7O. [0.5 mark] What is the value of state _ C G _ at depth 2? Note: the root node is at depth 0

a. -1

b. 0

c. 1 [Correct]

d. None of the above

7P. [0.5 mark] Which of the following action(s) should the first player take? Select all that is/are true.

a. _ C _ G [Correct]

b. C C G _ [Correct]

c. All actions result in losing the game

Question 8: Alpha-Beta Pruning

Consider the following minimax tree:

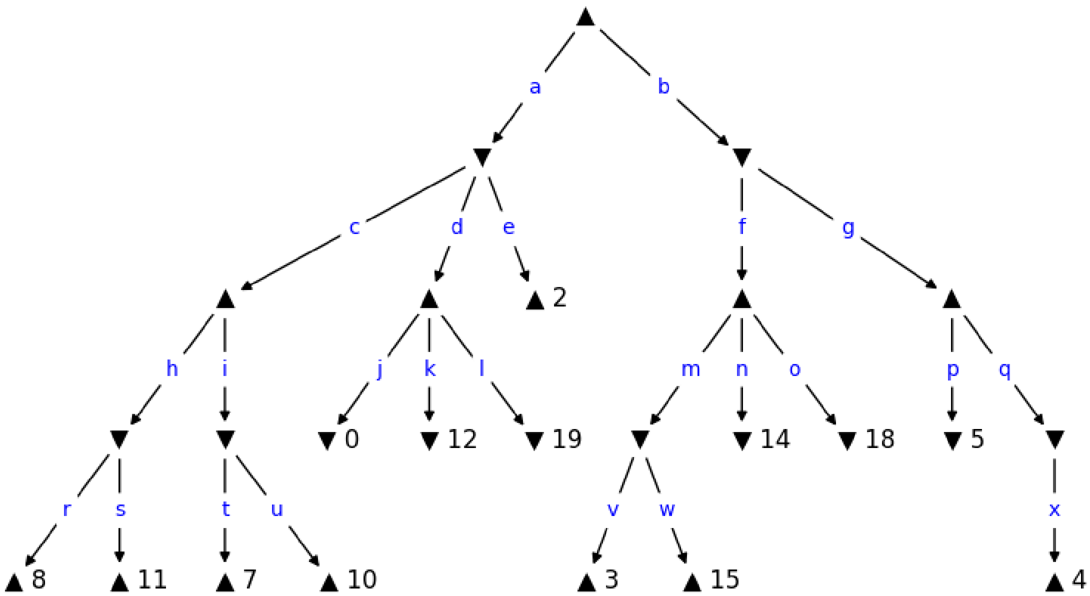


Figure 5: Minimax Tree

Note: The symbol ▲ represents the max player's turn, while ▼ indicates the min player's turn.

Answer question 8A

8A. [4 marks] Suppose we traverse this tree with DFS from **right-to-left**. Select **all** the link(s) that would be pruned by alpha-beta pruning algorithm. Select only the links that are **directly** pruned and not those that are indirectly pruned because they are in a subtree of a pruned link.

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

a. a

b. b

c. c [Correct]

d. d [Correct]

e. e

f. f

g. g

h. h

i. i

j. j

k. k

l. l

m. m [Correct]

n. n [Correct]

o. o

p. p

q. q

r. r

s. s

t. t

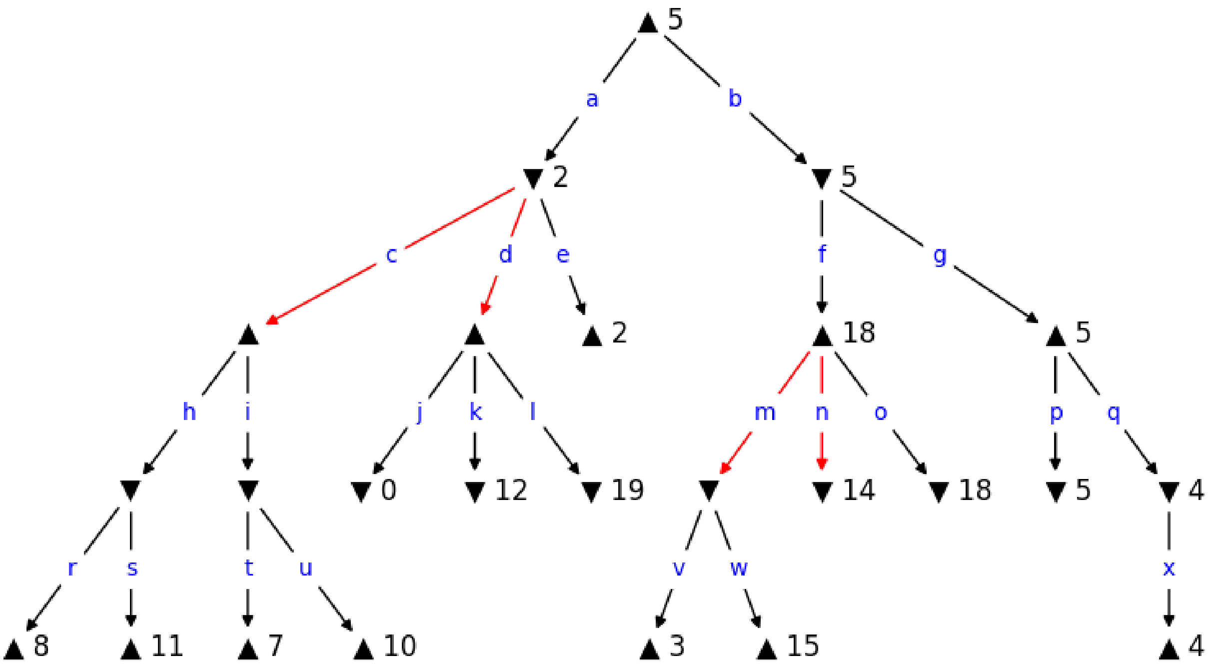
u. u

v. v

w. w

x. x

y. None of the above



— END OF PAPER —