COMP7015 Artificial Intelligence **Semester 1, 2024-2025**

Written Assignment 1

Instructor: Dr. Kejing Yin due on Sep. 27, 2024 (11:59am)

Submission instruction: *Put your answers to all questions to one single PDF file, make sure that your answers are readable. Name the file as "*`wa1_StudentID_YourFullName.pdf`*" and upload it to Moodle submission box.*

Problem 1 Formulating A Searching Problem (20 marks)

Consider a famous problem in AI, *the missionaries and cannibals problem*, which is stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. The task is to get everyone to the other side of the rivier, but you cannot leave a group of missionaries in one place outnumbered by the cannibals in the same place.

(a) (6 marks) Formulate the problem into a search problem. Define precisely its state space, the initial state, the goal state, and all possible actions.

Solution: The missionaries and cannibals problem can be formulated as follows:

(1) To define the **states**, we denote the number of missionaries, cannibals, and boat on the left side of the river by an array (m, c, b). Correspondingly, the number of missionaries, cannibals, and boat on the other side of the river are given by $3 - m$, $3 - c$, and $1 - b$.

(2) The **initial state** then can be denoted by $(3, 3, 1)$, i.e., three missionaries, three cannibals and one boat are on the left side of the river.

(3) The **goal state** is defined by $(0, 0, 0)$, i.e., all missionaries, cannibals and the boat are on the other side of the river.

(4) We use pq to denote an action of moving p missionaries and q cannibals across the river and use symbols $\triangleright$ and $\triangleleft$ to denote the direction of moving people. The **all possible actions can be given by**: $01\triangleright$, $02\triangleright$, $10\triangleright$, $11\triangleright$, $20\triangleright$, $01\triangleleft$, $02\triangleleft$, $10\triangleleft$, $11\triangleleft$, and $20\triangleleft$.

(b) (10 marks) Draw the complete state space according to your formulation.

Solution: The state space is given in Fig. 1.

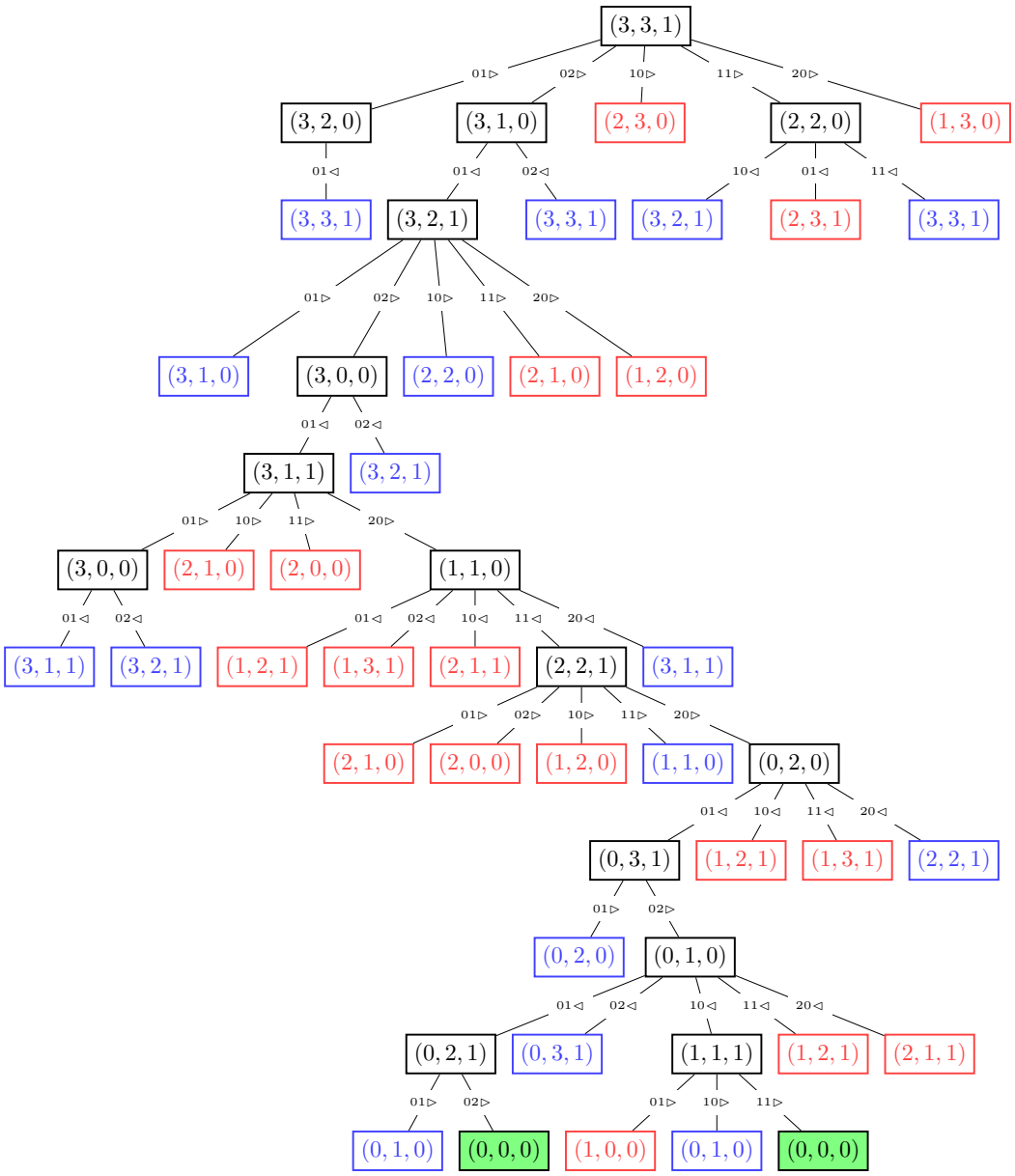


Figure 1: The complete state space of the missionaries and cannibals problem. The nodes in blue represent repeated states and that in red represent the states that violate the rules; therefore, they are not further expanded. The shaded in green are the goal states.

(c) (4 marks) Which search algorithm is the best one to solve for problem? Explain your choice.

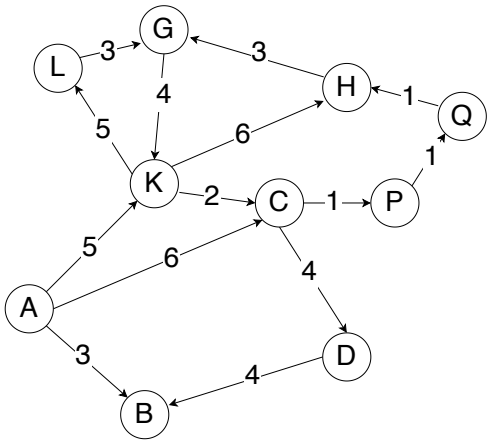
Solution: Either one of the following will be the correct answer:

- **BFS** or **uniform-cost search** as they both guarantee a cost-optimal solution.
- **Iterative deepening search** as it saves memory and guarantees to find a solution. Besides it is not required in this problem to find the cost-optimal solution, so even though it may not give the cost-optimal solution, it is still acceptable.

Remarks: you can define some notations to help you formulate the problem, like we did in the wolf, goat and cabbage problem. If you intend to use any self-defined notations, explain the meaning of your notations clearly and precisely at the begining of your solution.

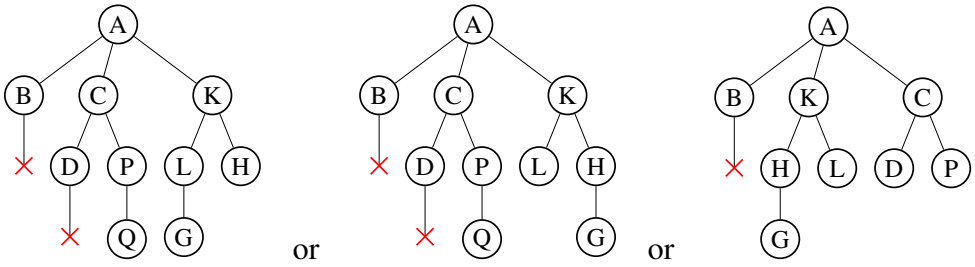
Problem 2 Uninformed Search (20 marks)

The following figure shows the flight connections between different cities. Each letter inside the circle represents a city, the arrow between two cities means that there is a flight between them and the direction of the arrow indicates the direction of the flight. The numbers above each arrow show the price of the flight tickets. Suppose you are running a travel agency and would like to use AI algorithms to find routes for your customers.



(a) (10 marks) A customer would like to find a flight from A to G. He/she does not care about the price and would like to have the minimum number of flight transfer. What algorithm should be used? Plot the search tree constructed by using this algorithm and write down the order of nodes that you expanded.

Solution: We use breadth-first search (BFS) as it finds the shallowest solution. In this problem, the flight with minimum transit corresponds to the shallowest solution. Some examples of correct search trees are:

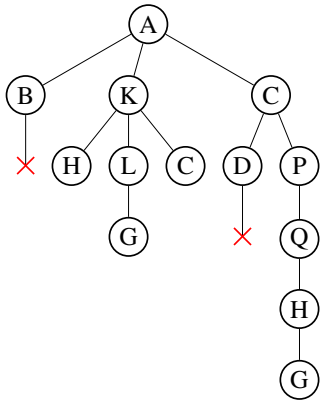


Some examples of correct orders of nodes expanded are (The answer must align with the search tree plotted):

- A, B, C, K, D, P, L, (G) *(check goal state when generating the node)*
- A, B, C, K, D, P, L, H, Q, (G) *(check goal state when expanding the node)*

(b) (10 marks) A customer would like to find the cheapest flight from A to G, no matter how many transits it needs. What algorithm should be used? Plot the search tree constructed by this algorithm and write down the order of nodes that you expanded.

Solution: We use uniform-cost search as it fins the cost-optimal solution. In this problem, the cheapest flight corresponds to the cost-optimal solution when we consider the price as the cost. The search tree constructed is:



The order of expanding nodes is:

- A, B, K, C, P, Q, H, D, L, G; *or*
- A, B, K, C, P, Q, H, L, D, G

Problem 3 Heuristic Search (40 marks)

Consider the following 8-puzzle problem.

7	2	4
5		6
8	3	1

Initial State

	1	2
3	4	5
6	7	8

Goal State

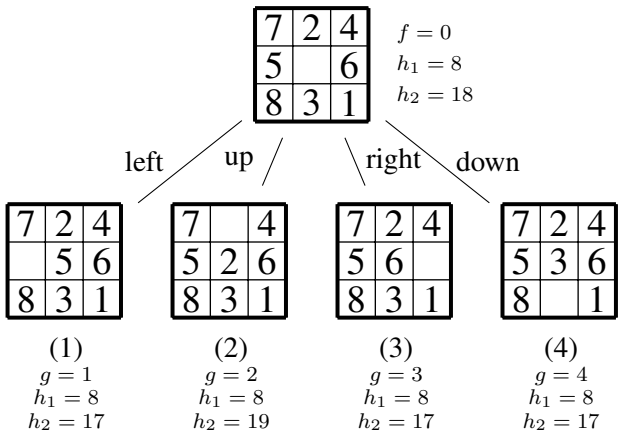
The objective of the puzzle is to slide the tiles horizontally or vertically into the blank space until the configuration matches the goal state. The search space is quite large in 8-puzzle or 16-puzzle

- h_1 = the number of misplaced tiles (blank not included).

  For the example above, all eight tiles are out of position, so the initial state has $h_1 = 8$.
- h_2 = the sum of the distances of the tiles from their goal positions.

  Because tiles cannot move along diagonals, the distance is the sum of the horizontal and vertical distances — called **Manhattan distance**. For the example above, moving 7 in the initial state to the correct location in the goal state requires 3 moves (right, down, and down), moving 2 in the initial state to the correct location in the goal state requires 1 move (right). Tiles 1 to 8 in the initial state of the above example give a Manhattan distance of $h_2 = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18$.

The possible actions include moving the blank tile left, up, right, and down. The cost of moving left, up, right, and down are 1, 2, 3, and 4, respectively. The first step of search expands the initial state and results in the following search tree:



(a) (6 marks) Are h_1 and h_2 admissible? Briefly explain your answer.

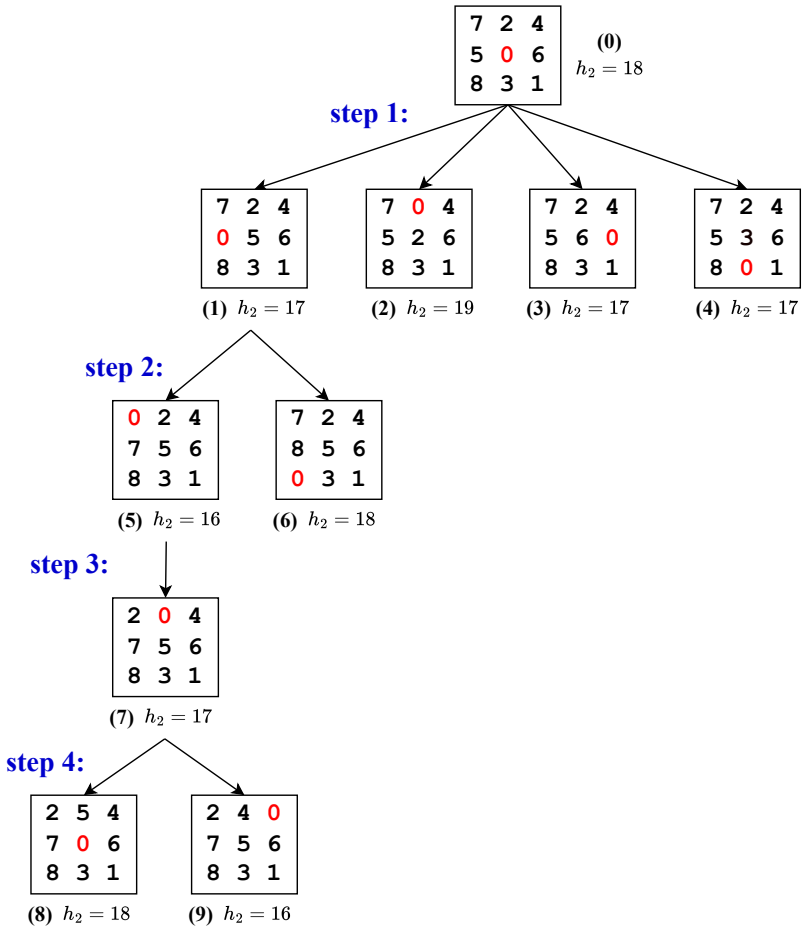
Solution: h_1 is admissible because for each misplaced number, it requires at least one step to recover it. So h_1 will not be larger than the actual cost for all states. h_2 is also admissible because for each misplaced tile, the minimum number to move it to the correct place is the Manhattan distance (since the tile can only move horizontally and vertically) between its current location and the goal location. Since there are other numbers blocking this shortest way, it will always take steps no less than the Manhattan distance. Therefore, h_2 will never overestimate the actual cost.

(b) (4 marks) Which of h_1 and h_2 is a better heuristic function to use? Briefly explain the reasons.

Solution: h_2 is a better heuristic function as it is a more accurate estimation of the cost from each state to the goal state. It is always grater than or equal to h_1 yet still admissible.

(c) (15 marks) Perform greedy search with the heuristic function h_2 to solve for this problem. Draw the portion of the search tree after the first four steps (including the first step shown above). Show which state you expand in each step and briefly explain the reason.

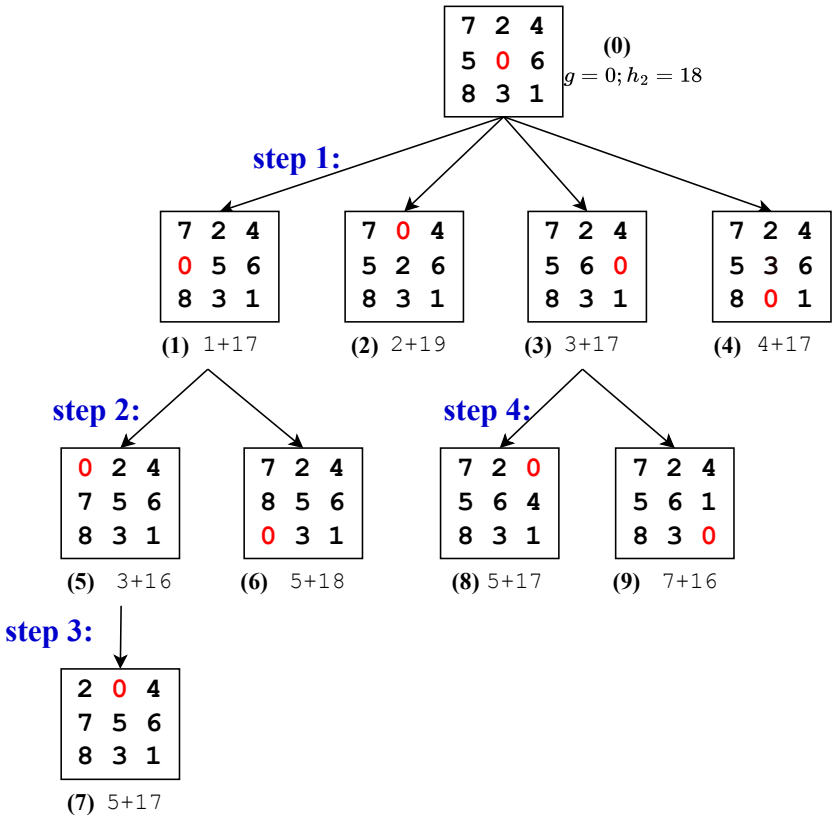
Solution: The resulting portion of the search tree is shown in the figure below.



In the first step, (0) is expanded because it is the only node in the frontiers. In the second step, (1), (3), and (4) have the smallest heuristic function h_2, so we randomly choose one to expand. In this example solution, we expand (1). In the third step, (5) is the node with smallest h_2, so we expand (5). In the fourth step, (3), (4), and (7) all have the same smallest h_2, so we randomly choose one. In this example solution, we expand (7).

(d) (15 marks) Perform A* search with the heuristic function h_2 to solve for this problem. Draw the portion of the search tree after the first four steps (including the first step shown above). Show which state you expand in each step and briefly explain the reason.

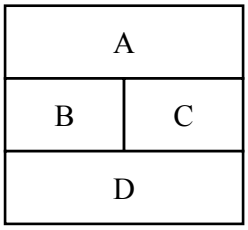
Solution: The resulting portion of the search tree is shown in figure below.



The numbers underneath each state are in the format of $g(n)+h_2(n)$, where g is the cost and h_2 is the heuristic function. In the first step, (0) is expanded because it is the only node in the frontiers. In the second step, (1) has the smallest $g + h_2 = 18$, so we expand (1). In the third step, (5) has the smallest $g + h_2 = 19$, so we expand (5). In the fourth step, (3) has the smallest $g + h_2 = 20$, so we expand (3).

Problem 4 Constraint Satisfaction Problem (20 marks)

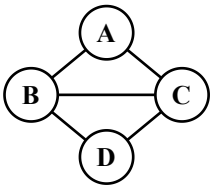
Given the following map with four regions, we would like to color each region using one color from red, green, or blue. Adjacent regions cannot have the same color. Answer the following questions.



(a) (5 marks) Formulate this as a constraint satisfaction problem, define its variables, domains, and constraints. Draw a constraint graph to represent the problem.

Solution: We define four variables, denoted by A, B, C, D, corresponding to the four regions in the figure. Their domains are $A \in \{\text{red}, \text{blue}, \text{green}\}$; $B \in \{\text{red}, \text{blue}, \text{green}\}$; $C \in \{\text{red}, \text{blue}, \text{green}\}$; and $D \in \{\text{red}, \text{blue}, \text{green}\}$. The con-

straints are $A \neq B$, $A \neq C$, $B \neq C$, $B \neq D$, and $C \neq D$. The constraint graph is given by:



(b) (3 marks) Give out one solution to this constraint satisfaction problem.

Solution: A possible solution is $A =$ red, $B =$ blue, $C =$ green, $D =$ red.

(c) (8 marks) Can we color A as red and D as blue? Show the answers using arc consistency with detailed steps.

Solution: No, we cannot. This can be shown using arc consistency. The detailed steps are as below. We use "r", "g", "b" to represent "red", "green", and "blue" respectively.

1. The initial domains are: $A \in \{r, g, b\}; B \in \{r, g, b\}; C \in \{r, g, b\}, D \in \{r, g, b\}$.
2. Assign $A = r$, remove g and b from the domain of A: $A \in \{r\}; B \in \{r, g, b\}; C \in \{r, g, b\}, D \in \{r, g, b\}$.
3. Enforce arc consistency between (A, B): r is removed from domains of B: $A \in \{r\}; B \in \{g, b\}; C \in \{r, g, b\}, D \in \{r, g, b\}$.
4. Enforce arc consistency between (A, C): r is removed from domains of C: $A \in \{r\}; B \in \{g, b\}; C \in \{g, b\}, D \in \{r, g, b\}$. All other arcs are consistent.
5. Assign $D = b$, remove r and g from the domain of D: $A \in \{r\}; B \in \{g, b\}; C \in \{g, b\}, D \in \{b\}$.
6. Enforce arc consistency between (B, D): b is removed from the domain of B: $A \in \{r\}; B \in \{g\}; C \in \{g, b\}, D \in \{b\}$.
7. Enforce arc consistency between (C, D): b is removed from the domain of C: $A \in \{r\}; B \in \{g\}; C \in \{g\}, D \in \{b\}$.
8. Enforce arc consistency between (B, C): b is removed from the domain of B: $A \in \{r\}; B \in \emptyset; C \in \{g\}, D \in \{b\}$. The domain of B is empty. The partial assignment fails.

(d) (4 marks) Can we obtain the same results as showed in the answers to subquestion (c) if we use forward checking?

Solution: No, we cannot. In forward checking, after assigning A=r and D=b, the domain of A, B, C, and D are not empty. Therefore, the forward checking cannot detect that this assignment violates the rules.