Northeastern University
CS5100 – Foundations of Artificial Intelligence
Fall 2022 Semester, Richard Hoshino

Problem Set #1

This Problem Set is due at **6PM on Wednesday September 21**, and will be submitted on Canvas.

This Problem Set will be marked out of 40, and is worth 8% of the final course grade. There are five problems, each worth eight marks.

Problems #1 and #2 and #3 are to be completed individually, while Problems #4 and #5 are to be completed in your assigned small groups. For the two group problems, you are strongly encouraged, though not required, to have a different lead author for each of parts (a), (b), and (c).

Please type (or neatly handwrite) your solutions on standard 8.5×11 paper, with your name(s) at the top of each solution. Ensure that you submit five *separate* PDF files on Canvas, one for each problem. Make sure you label your Problem Set #1 submissions appropriately - e.g. richard1-1.pdf, richard1-2.pdf, richard1-3.pdf, richard1-4.pdf, richard1-5.pdf.

Given that the last two problems are done in a group, your final two PDF files will be identical to some of your classmates. (For example, richard1-4.pdf might be identical to yvonne1-4.pdf and bethany1-4.pdf). This is completely fine, and enables you to have a record of all of your submitted work in this course.

While a solution must be absolutely perfect to receive full marks, I will be generous in awarding partial marks for incomplete solutions that demonstrate progress.

So that there is no ambiguity, there are two non-negotiable rules. A violation of either rule constitutes plagiarism and will result in you receiving an F for this course.

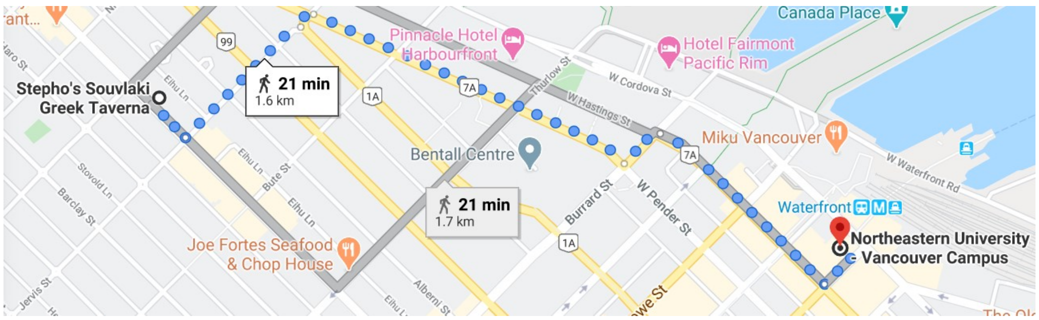
(a) If you meet with a classmate to discuss one of the three Individual Problems, the articulation of your thought process (i.e., what you submit to me), must be an individual activity, done in your own words, away from others. Please remember that the solution-writing process is where so much of your learning will occur in this course: much more than anything we do in class, and even more than the time you spend on solving the problems. Do not be surprised if it takes you 3 to 5 times as long to write up a solution than it takes you to actually solve the problem. (For me, as an academic researcher writing formal proofs for publications, my ratio is significantly higher!)

(b) This Problem Set has been designed to be challenging, because struggling through problems is how we learn best. Your educational experience is cheapened by going online and finding the solution to a problem; even using the Internet to look for a "small hint" is unacceptable. In return, I will be readily available during our optional problem-solving workshops on Tuesday morning, and upon request, I will post hints to any questions you have on our class Canvas Page.

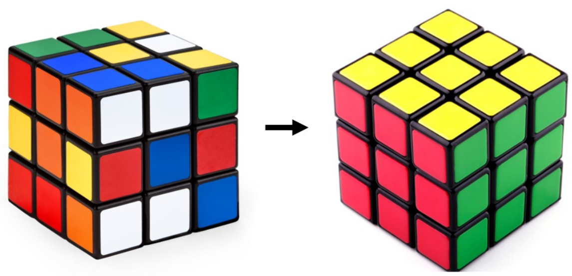
Problem #1 – INDIVIDUAL

Clearly explain why each of the following problems are actually AI search problems. To do this, identify the initial state, actions, transition model (or successor function), goal test, and path cost. Write one or two sentences for each of these five components.

(a) Determining the shortest path from your favourite restaurant in Vancouver to the Northeastern Vancouver campus.



(b) Solving a jumbled $3 \times 3 \times 3$ Rubik's Cube in the minimum possible number of moves.



(c) Completing a (partially-filled) Sudoku puzzle and obtaining the (unique) solution.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

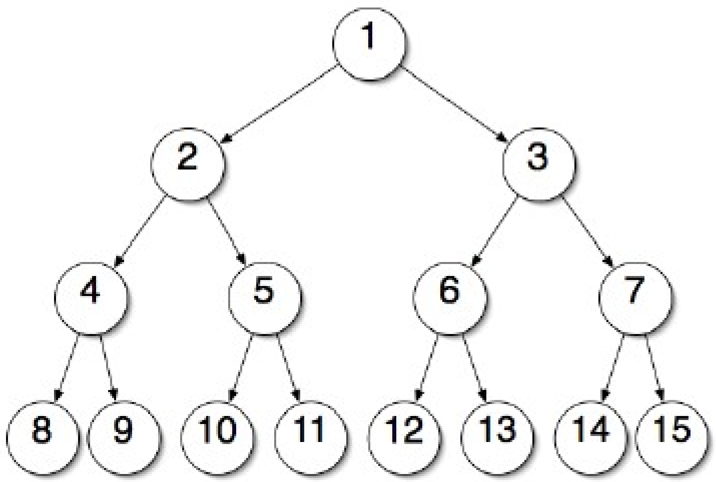
→

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Problem #2 – INDIVIDUAL

Consider a state space, where each state is a positive integer. The start state is 1. For all positive integers $k \geq 1$, state k has two successors: $2k$ (Left) and $2k+1$ (Right).

This state space can be represented as a graph. The picture below shows states at Level 0, Level 1, Level 2, and Level 3.



(a) Suppose the goal state is 18. List the order in which the states (or nodes) will be visited for breadth-first search, and for depth-limited search with depth limit 4. Will depth-first search ever find this goal state? Explain.

(b) Suppose the goal state is n. If n is a very large number, breadth-first search and depth-limited search are inefficient. Find a fast algorithm that quickly generates the sequence of Left and Right steps needed to reach this goal state. Explain how your algorithm works, and use your algorithm to generate the sequence of Left and Right moves that reaches the goal state of 2022. Finally, determine the running time of your algorithm (e.g. $O(n^2)$, $O(n)$, $O(\log n)$, etc.).

(c) For each number n, let $f(n)$ be the binary representation of n. For example, $f(10) = 1010$ and $f(63) = 111111$. Explain how the binary representation $f(n)$ tells you how to generate the sequence of Left and Right steps needed to reach goal state n. Apply your explanation to the goal state $n = 2022$, and confirm your answer is the same as your sequence in part (b).

Problem #3 – INDIVIDUAL

In this question, you will explore the AI Search game Mastermind, a popular board game from the 1970s. For an online version of Mastermind, check out

`http://www.onlinespiele-sammlung.de/mastermind/mastermindgames/madglibs/index.html`

Your goal is to determine the computer's hidden code. You know that the code consists of four *distinct* colours chosen from the set {A,B,C,D,E,F}. For example, EFBC is a valid code, but EFBF and EEEE are not. There are $6 \times 5 \times 4 \times 3 = 360$ possible codewords.

In each turn, you guess the hidden code. Note that each of your guesses must be a possible code: thus EFBC is a valid guess, but EFBF and EEEE are not.

After you make a guess, the computer compares your guess to the hidden code, and displays a red peg for each colour in the correct position, and a white peg for each colour in the wrong position.

For example, if EFBC is the hidden code and your guess is ABCD, then the computer will display two white pegs. However, if your guess is ABDC, then the computer will display one red peg and one white peg.

The game is complete when the computer displays four red pegs, confirming that your final guess matches the hidden code.

(a) On your first turn, you guess ABCD, after which the computer displays two red pegs and two white pegs. On your second turn, you guess ABEF, after which the computer displays two white pegs. Determine what the hidden code must be, clearly justifying your answer.

(b) For some integer n of your choice, determine an algorithm that correctly identifies the four distinct colours of the hidden code in at most n guesses. Clearly explain how your algorithm works, and why your algorithm is guaranteed to identify the four colours in at most n guesses, no matter what the hidden code is. You will receive full marks for an algorithm with $n \leq 5$ and partial marks for an algorithm with $n \geq 6$.

(c) For some integer m of your choice, determine an algorithm that correctly identifies the hidden code in at most m guesses. Clearly explain how your algorithm works, and why your algorithm is guaranteed to identify the hidden code in at most m guesses, no matter what the hidden code is. You will receive full marks for an algorithm with $m \leq 10$ and partial marks for an algorithm with $m \geq 11$.

Problem #4 – GROUP

AI researchers have created various chess-playing robots, i.e., robots that have been programmed to play a competitive game of chess against human players. For one such example, check out this two-minute video on "Raspberry Turk":

`https://www.youtube.com/watch?v=lpJIBVU_WJE`

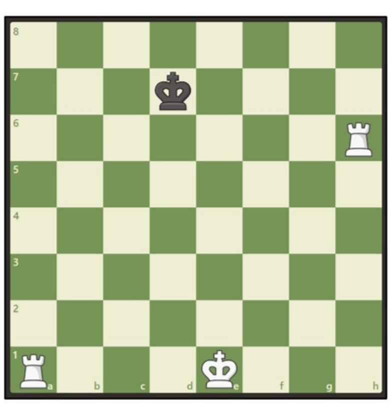
(a) Determine the *task environment* of a chess-playing robot by specifying the performance measure, environment, actuators, and sensors. Also describe the properties of this task environment, by explaining whether it is fully or partially observable, single or multi agent, deterministic or stochastic, episodic or sequential, static or dynamic, discrete or continuous, known or unknown. Write one or two sentences for each response. (Some questions have more than one possible response. As long as you justify your response, you will receive full credit.)

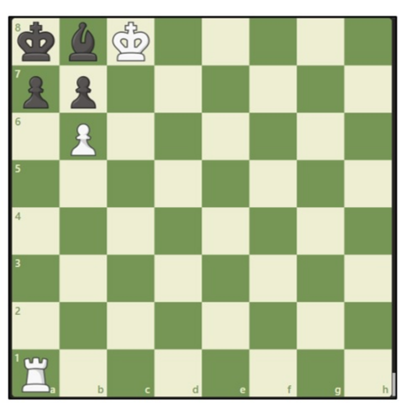
(b) A knight is placed on the top-left corner of an 8×8 chessboard, indicated by square 1. Determine the minimum number of "knight moves" needed to move the knight from the top-left corner of the board (square 1) to the bottom-right corner (square 64). Clearly justify your answer.

Let $f(n)$ be the minimum number of knight moves required to move a knight from the top-left corner to the bottom-right corner of an $n \times n$ chessboard. You determined the value of $f(8)$ above. Now determine a general formula for $f(n)$ and prove that your formula is correct.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

(c) In both of these chess configurations, you are White, and it is White's turn to move. For each of these problems, your goal is to achieve Checkmate in the *minimum* possible number of moves. In your solution, clearly explain how your sequence of moves guarantees Checkmate, no matter how hard Black tries to avoid losing. You will receive full points for an optimal solution, and partial points for a sub-optimal solution.



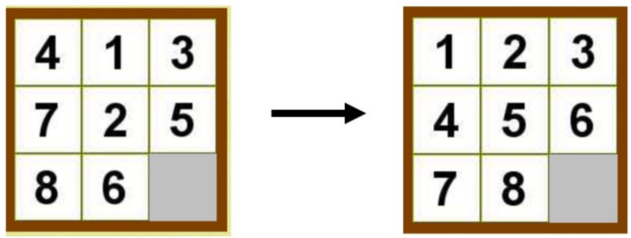


Problem #5 – GROUP

In this question, you will explore the 8-puzzle, a popular sliding block game from the 1880s. For an online version of the 8-puzzle, check out

`http://www.tilepuzzles.com/default.asp?p=12`

The object of the puzzle is to place the tiles in order by making sliding moves that use the empty space. The desired goal state is indicated on the right.



You will explain how the 8-puzzle can be efficiently solved using the A* search algorithm.

(a) Let h_1 be the heuristic function corresponding to the number of misplaced tiles, and let h_2 be the heuristic function corresponding to the sum of the "Manhattan distance" of the tiles from their goal positions. In the above problem, $h_1 = 7$, since all the tiles other than 3 are out of position, and $h_2 = 1+1+0+1+1+2+1+1 = 8$, since six of the seven misplaced tiles are one move away from their destination, while Tile #6 is two moves away.

For each of these two heuristic functions, solve the above puzzle in as few moves as possible, using the A* search algorithm. Make sure you clearly show your steps, drawing out the two search trees and listing out the steps in your optimal solutions.

(b) For each of these two heuristic functions, h_1 and h_2, prove that it is both admissible and consistent. In general, is h_1 or h_2 a better heuristic function for the 8-puzzle problem? Clearly justify your answer.

(c) For each game state, list the eight numbers in order, from top left to bottom right. (In the above problem, the initial state is 41372586). For any given state, we define the following "score": count one point for each pair (i, j), with $1 \leq i < j \leq 8$, for which j appears before i in the initial state.

In the above problem, the score of the initial state is 8, as the following pairs appear in the wrong order: $\{(1,4), (3,4), (2,4), (2,3), (2,7), (5,7), (6,7), (6,8)\}$.

Prove that the initial state 41372586 is solvable, and the initial state 41372568 is not solvable. (Hint: what do you notice about the scores of all states that are solvable? Can you explain why?)

There are $8! = 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 40320$ different ways to create an initial state for the 8-puzzle, with the empty space in the bottom-right corner. Of these 40320 initial states, determine how many are solvable. Clearly justify your answer.