

CSE 486 – Introduction to Artificial Intelligence:

Assignment #1

Due on 15 September 2023 at 11:59pm

Dr Khodakhast Bibak Section A

Emil Sayahi

Problem 1

(0 points) Optional.

Problem 2

(8 points) Pacman and Ms. Pacman are lost in an $N \times N$ maze and would like to meet; *they don't care where*.

In each time step, *both* simultaneously move in one of the following directions: {NORTH, SOUTH, EAST, WEST, STOP}.

They do *not* alternate turns. We must devise a plan which positions them together, somewhere, in as few time steps as possible. Passing each other does not count as meeting; they must occupy the same square at the same time. We can formally state this problem as a *single-agent* state-space search problem as follows:

States: The set of pairs of positions for Pacman and Ms. Pacman, that is, $\{(x_1, y_1), (x_2, y_2) \mid x_1, x_2, y_1, y_2 \in \{1, 2, \dots, N\}\}$

Size of state space: N^2 for both Pacmen, hence N^4 total

Goal test: $\text{Goal}((x_1, y_1), (x_2, y_2)) := (x_1 = x_2) \wedge (y_1 = y_2)$

Part A

(2 points) Determine the branching factor.

Solution

Each has a choice of five actions; $5 \cdot 5 = 25$

Part B

(6 points) For each of the following graph search methods determine and explain whether it is guaranteed to output optimal solutions to *this* problem?

1. DFS
2. BFS
3. UCS

Solution

BFS and UCS are both optimal for this problem; with BFS, it is optimal when the edges have the same or no cost, while UCS is itself an optimal algorithm. DFS is not optimal, as it is not an optimal algorithm in any case.

Problem 3

(6 points) Consider a state space where the start state is number 1 and each state k has two successors: $2k$ and $2k + 1$. Suppose you want to navigate your robot from state 1 to state 2020. Can you find an algorithm that outputs the solution to this problem without any search at all? Output the solution for 2020.

Solution

The state space is a binary tree, where the parent of k is $\lfloor \frac{k}{2} \rfloor$. Therefore, to obtain the path from 2020 to 1, it is possible to successively calculate parent nodes until we reach 1. The result of these calculations is the path from 2020 to 1 being $2020 \rightarrow 1010 \rightarrow 505 \rightarrow 252 \rightarrow 126 \rightarrow 63 \rightarrow 31 \rightarrow 15 \rightarrow 7 \rightarrow 3 \rightarrow 1$. The path from 1 to 2020 would be the reverse of this path— $1 \rightarrow 3 \rightarrow 7 \rightarrow 15 \rightarrow 31 \rightarrow 63 \rightarrow 126 \rightarrow 252 \rightarrow 505 \rightarrow 1010 \rightarrow 2020$.

Problem 4

(4 points) Your goal is to navigate a robot out of a maze. The robot starts in the center of the maze facing north. You can turn the robot to face north, east, south, or west. You can direct the robot to move forward a certain distance, although it will stop before hitting a wall. We can formulate this problem as follows. We'll define the coordinate system so that the center of the maze is at $(0,0)$, and the maze itself is a square from $(-1,-1)$ to $(1,1)$.

Initial state: Robot at coordinate $(0,0)$, facing North.

Successor function: Move forwards any distance d ; change direction robot is facing.

Cost function: Total distance moved.

Part A

(2 points) Let (x,y) be the current location. What is the Goal test?

Solution

The goal test is $(x,y) \in \{(-1,-1), (-1,1), (1,-1), (1,1)\}$; ie, the robot is at one of the corners of the maze, when $(|x|, |y|) = (1, 1)$.

Part B

(2 points) How large is the state space?

Solution

The state space has an infinite number of states, because the robot can move any distance d in any direction, and the distance d can be any real number (ie, $d \in \mathbb{R}$). Any state can therefore be described as $(x \in \mathbb{R}, y \in \mathbb{R})$, where $-1 \leq x \leq 1$ and $-1 \leq y \leq 1$ while the robot is still in the maze.

Problem 5

(6 points) For each of the following assertions, say whether it is true or false. Justify your answers.

Part A

(2 points) Imagine the next Mars rover malfunctions upon arrival on Mars. From this we can deduce that Mars rover is not a rational agent. (Note that a rational agent is not necessarily perfect, it's only expected to maximize goal achievement, given the available information.)

Solution

False. The rover might've been hit by an object causing the malfunction, for example, despite having an ideal design.

Part B

(2 points) Every optimal search strategy is necessarily complete.

Solution

True. An algorithm is optimal if it always finds the best solution when one exists; an algorithm is complete if it always finds *some* solution when one exists.

Part C

(2 points) Breadth-first search is optimal if the step cost is positive.

Solution

False. Breadth-first search is optimal when all edges have the same cost, no cost, or when the cost is a non-decreasing function of the depth of the node.

Problem 6

(6 points) For the search problem shown below, S is the start state, and G is the (only) goal state. Break any ties alphabetically. What paths would the following search algorithms return for this search problem? Give your answers in the form ‘S - A - D - G’.

Part A

(2 points) BFS

Solution

$S \rightarrow B \rightarrow G$

Part B

(2 points) UCS

Solution

$S \rightarrow A \rightarrow D \rightarrow G$

Part C

(2 points) DFS

Solution

$S \rightarrow A \rightarrow C \rightarrow G$