CS50AI 2020 - Quiz 0 Submission

A score of 70% or higher is required to be considered to have "passed" a quiz.

Unlike CS50x, assignments and quizzes in this course are graded on a set schedule, and depending on when you submitted, it may take up to three weeks for your work to be graded. Do be patient! Quizzes (which are submitted via Google Forms and not submit50) will not show up as submitted in your Gradebook until the scores have been released; therefore, we ask that you please do not attempt to submit multiple times before your score is released.

* Erforderlich

E-Mail-Adresse *

yannique.hecht@ascentize.com

Name *

Yannique Hecht

edX Username *

Yannique

GitHub Username *

Meine Antwort

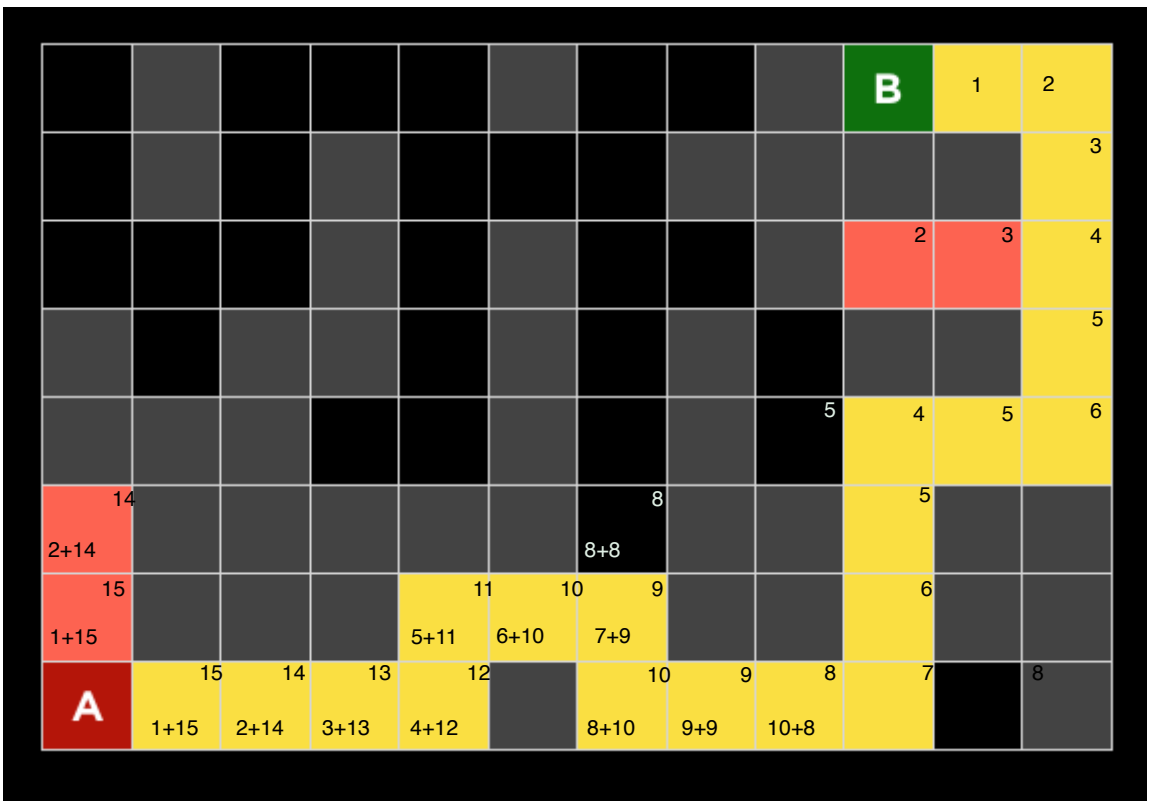
City, State, Country

Meine Antwort

Between depth first search (DFS) and breadth first search (BFS), which will find a shorter path through a maze? * 1 Punkt

- ○ DFS will always find a shorter path than BFS
- ○ BFS will always find a shorter path than DFS
- ○ DFS will sometimes, but not always, find a shorter path than BFS
- ○ BFS will sometimes, but not always, find a shorter path than DFS
- ○ Both algorithms will always find paths of the same length

The following question will ask you about the below maze. Grey cells indicate walls. A search algorithm was run on this maze, and found the yellow highlighted path from point A to B. In doing so, the red highlighted cells were the states explored but that did not lead to the goal.



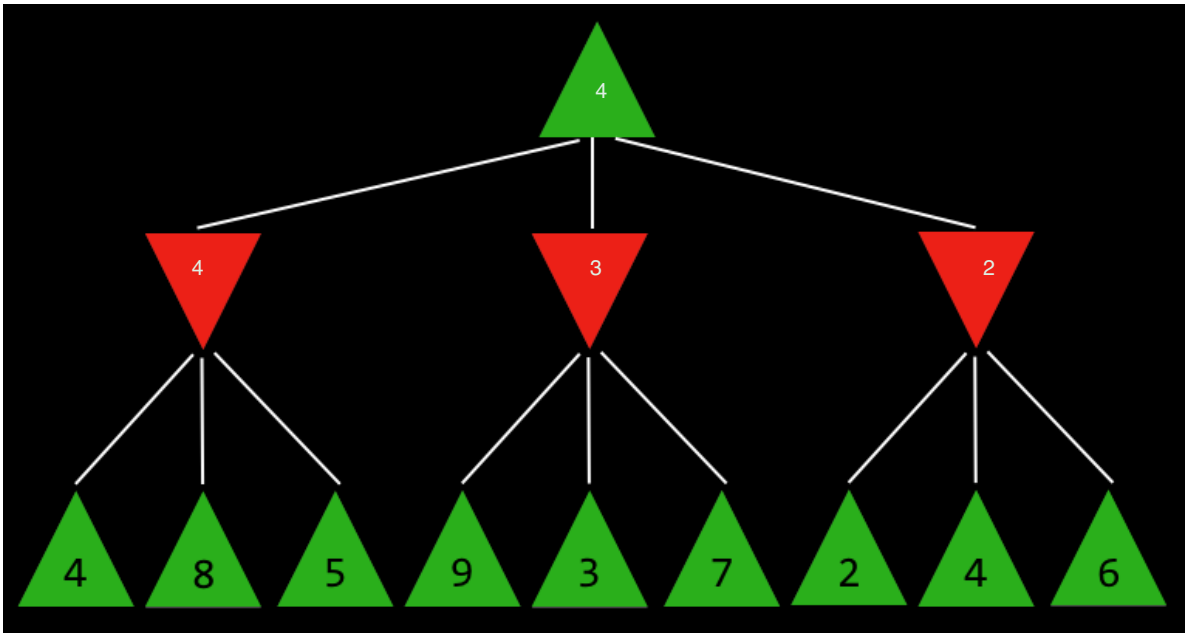
Of the four search algorithms discussed in lecture — depth-first search, breadth-first search, greedy best-first search with Manhattan distance heuristic, and A* search with Manhattan distance heuristic — which one (or multiple, if multiple are possible) could be the algorithm used? * 1 Punkt

- ~~Could only be A*~~
- Could only be ~~greedy best-first search~~
- ~~Could only be DFS~~
- ~~Could only be BFS~~
- ~~Could be either A* or greedy best-first search~~ (selected)
- ~~Could be either DFS or BFS~~
- ~~Could be any of the four algorithms~~
- Could not be any of the four algorithms

Why is depth-limited minimax sometimes preferable to minimax without a depth limit? * 1 Punkt

- Depth-limited minimax can arrive at a decision more quickly because it explores fewer states
- Depth-limited minimax will achieve the same output as minimax without a depth limit, but can sometimes use less memory
- Depth-limited minimax can make a more optimal decision by not exploring states known to be suboptimal
- Depth-limited minimax is never preferable to minimax without a depth limit

The following question will ask you about the Minimax tree below, where the green up arrows indicate the MAX player and red down arrows indicate the MIN player. The leaf nodes are each labelled with their value.



What is the value of the root node? * 1 Punkt

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

Comments, if any

Meine Antwort

Sie erhalten unter der von Ihnen angegebenen E-Mail-Adresse eine Kopie Ihrer Antworten.

Senden

Geben Sie niemals Passwörter über Google Formulare weiter.