Artificial Intelligence: Answers - Week 2 Search

Birna van Riemsdijk, Mannes Poel

Bart Stam

Contents

Question 1 **3**

Question 2 **3**

Question 3 **3**

Question 4 **5**

Question 5 **5**

Question 6 **7**

Question 7 **7**

Introduction

In this document one can find the answers and explanation to the AI tutorial exercises on search of week 2. Solutions and explanations are provided by Bart Stam and Birna van Riemsdijk. But any feedback or additions are more than welcome.

Question 1

Correct answer: B

Reasoning for (i)
Search agents need a complete model of the environment and the effect of the agent's actions. A simple reflex agent does not have the complexity to model this. The actions are solely based on its current perception. No notion of goal, state model, or transition function is present in a simple reflex agent. Therefore, search is not programmable on a simple reflex agent.

Reasoning for (ii)
Goal based agents can evaluate how likely it is for a node to bring them closer to the goal. This can allow them to "search" for the goal node. Thus, search is programmable on goal based agents.

Question 2

Correct answer: B

(i) Informed search strategies use a heuristic to prune the search space, while uninformed (blind) search strategies explore the state space without any information about which path is more likely to be the shortest path towards the goal.
(ii) Uninformed search strategies may not yield the optimal path towards the goal. Consider depth-first search. Depending on the order in which nodes are added to the frontier upon expansion, different path may be followed. Since no information about the length of the path informs the choice of nodes to expand, the path that is found may not be optimal.

Question 3

Correct answer: B

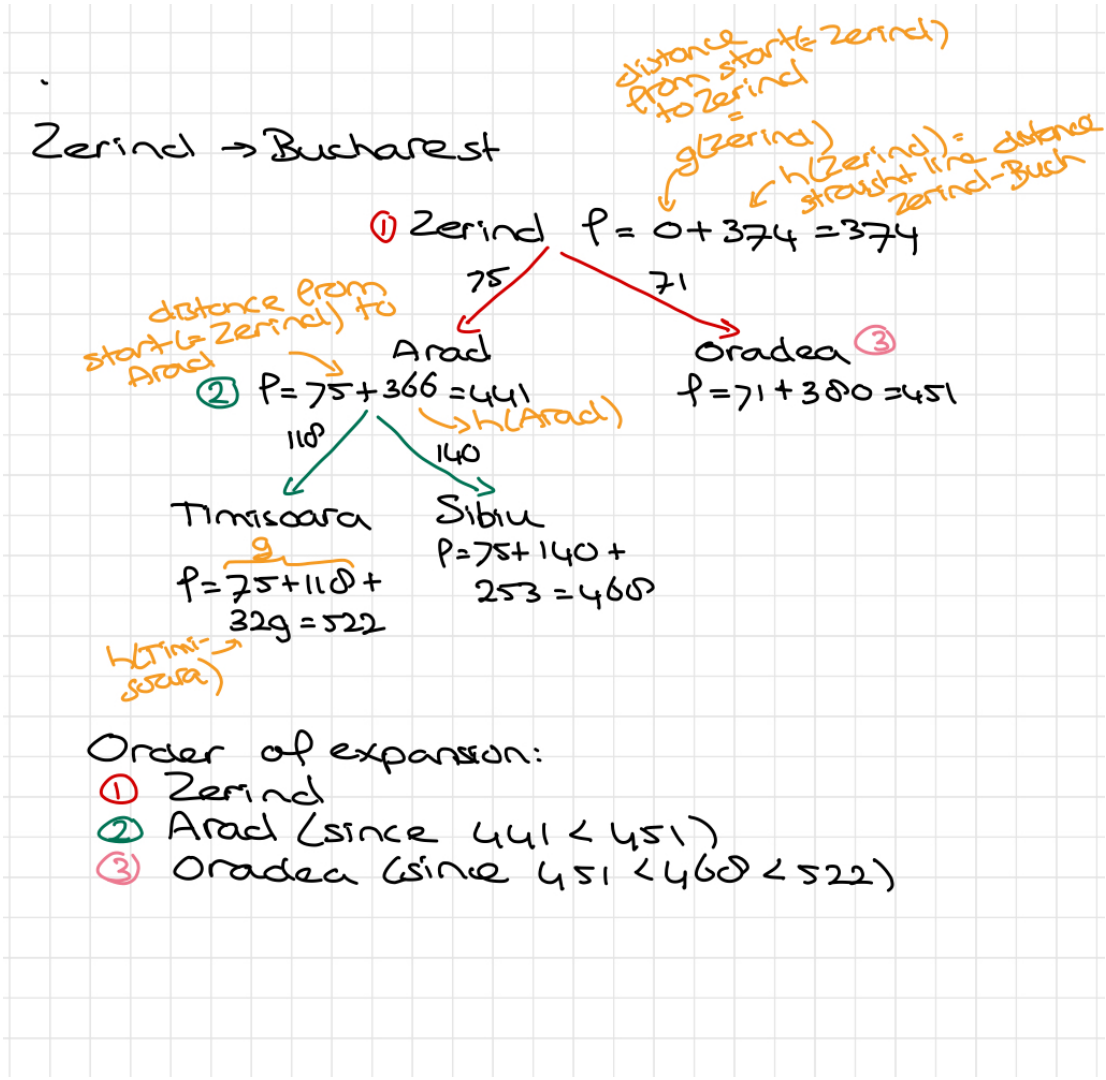
Reasoning for (i)
Depth First Search makes use of a so-called "closed list" that keeps track of what nodes have already been visited. The algorithm will not visit these nodes more than once. Depth First *Tree* Search does not maintain this closed list, and so it could get stuck in an infinite loop.

Depth First (Tree) Search works by expanding one path as far as possible until it either reaches the goal node or cannot go any further. If it chooses to expand an infinite branch of the graph before having found a possible solution, it will keep going and never terminate.

Reasoning for (ii)
When the Depth First (Tree) Search algorithm finds a path to the target node, it will never expand any other branch further than the length of the path it has already found to the goal. For example, if it finds a path of length 7 to the goal node, the maximum depth it will expand any other branch is 7. Therefore, if the algorithm finds a possible shortest path before expanding an infinite branch or loop, it will terminate even if there are cycles or infinite branches.

Question 4

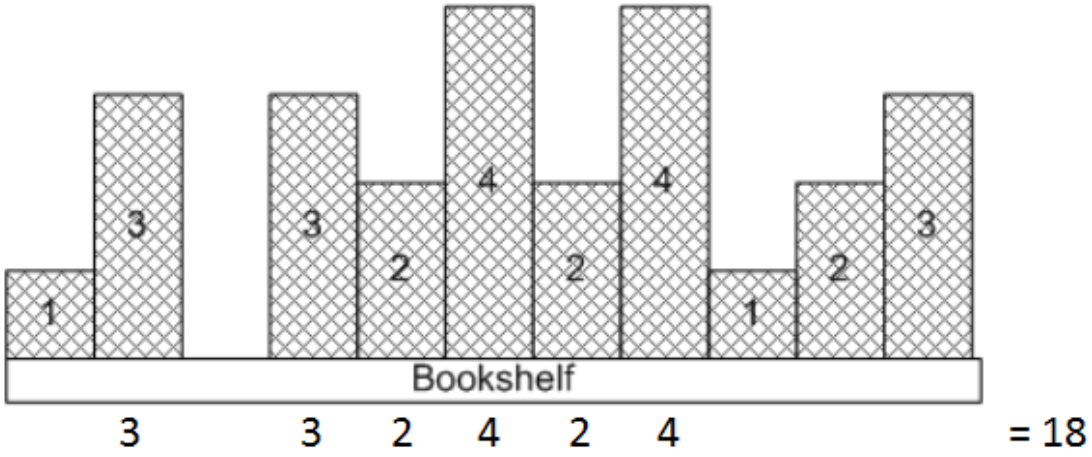
Correct answer: A



The picture above shows the search tree and the calculated f values for each node. First Zerind is expanded. Since the node Arad has the lowest f value among the leaf nodes of the tree (Arad and Oradea), it is expanded next. Of the remaining nodes in the frontier, i.e., the leaf nodes in the tree, Oradea now has the lowest f value since it is lower than that of Timisoara and Sibiu. Therefore Oradea is expanded next.

Question 5

Correct answer: A



A heuristic is a way to estimate the quality of a state. In this case, a state is a configuration of books on the bookshelf. The heuristic we use is: for every book, if there is a lighter book to the right of it, we add its weight to our estimation of the state. Above is a picture indicating how we estimate the quality of our current bookshelf state using said heuristic.

Question 6

Correct answer: D

The A* algorithm chooses which nodes to explore first based on the heuristic of the node *plus the cost it takes to get there*. In this case, the lower this value, the better.

$h([1,0,3,3,2,4,2,4,1,2,3]) + (costofmove) = 18 + 3 = 21$
$h([1,3,3,0,2,4,2,4,1,2,3]) + (costofmove) = 18 + 3 = 21$
$h([1,3,2,3,0,4,2,4,1,2,3]) + (costofmove) = 18 + 4 = 21$
$h([0,3,1,3,2,4,2,4,1,2,3]) + (costofmove) = 18 + 2 = 20$

As we can see, the quality of the new bookshelf states is 18 for each of the options. Doing the move in option 4 is the cheapest, so the A* algorithm will choose to expand that one first.

Question 7

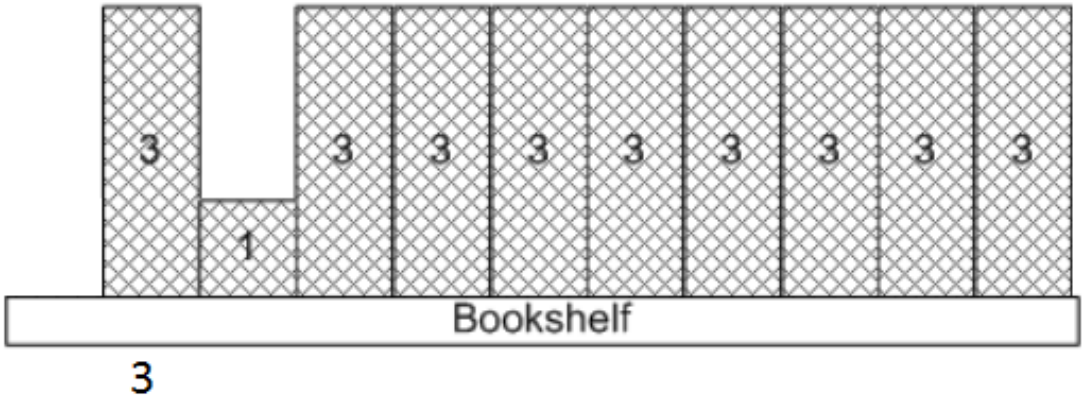
Correct answer: D

Definition of admissible
A heuristic is said to be admissible if it never overestimates the cost of reaching the goal. In the bookshelf scenario, this would mean that the cost (in terms of the two possible moves) to order the entire bookshelf starting at any state is never more than the estimated quality of that state given our heuristic.

Definition of consistent
A heuristic is said to be consistent if its estimate is always less than or equal to the estimated distance from any neighboring vertex to the goal, plus the step cost of reaching that neighbor. In other words, doing a step (like moving a book) never improves the (estimated) quality of the state by more than the cost of the step.

Reasoning
From the book (and lectures): a consistent heuristic is *always* admissible. For that reason, a heuristic that is not admissible can never be consistent.

The easiest way to prove that a heuristic is neither admissible nor consistent is to give an example of a scenario where it is not admissible. Below is an example of such a scenario.



Imagine the following bookshelf. As we can see, the heuristic estimates the quality of this state to be 3. However, we can reach the goal state by simply moving the book with cost 1 over the book to its left. This move has a cost of 2. This means that our heuristic overestimated the cost of reaching the goal. Thus, it is not admissible.

This is also a good example of why this heuristic is not consistent, since a move with cost 2 improves the quality of the state by more than 2.