CSPB 3202 - Truong - Artificial Intelligence

Started on	Monday, 20 May 2024, 3:00 PM
State	Finished
Completed on	Friday, 24 May 2024, 5:00 PM
Time taken	4 days 2 hours
Marks	9.00/9.00
Grade	**10.00** out of 10.00 (**100**%)

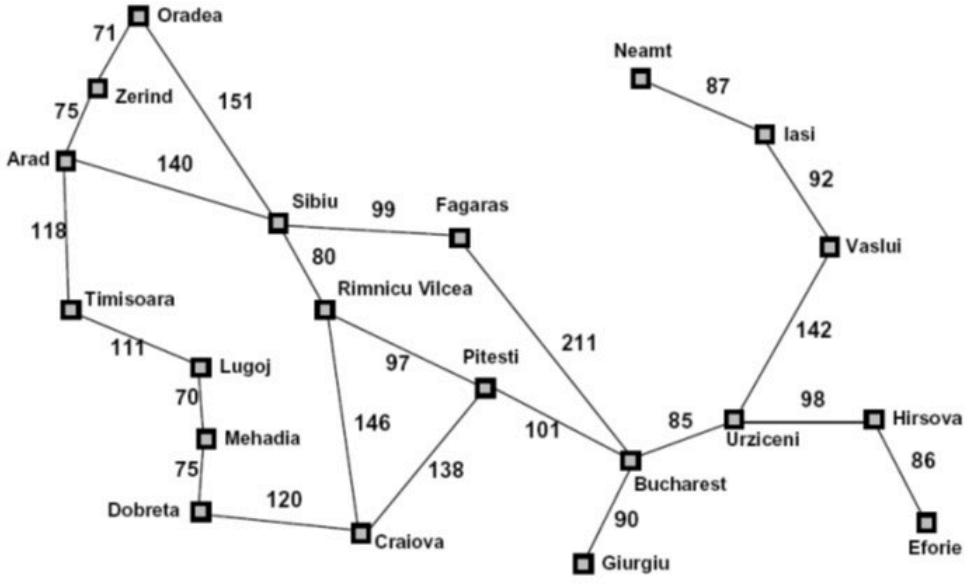
Question **1**

Correct

Mark 3.00 out of 3.00

Suppose you use **greedy (best-first) search** to find a path from state Zerind to Bucharest.

In what order are the states **explored**? (Note that "states" is being used synonymously with "nodes" here - these are actually cities in Romania.)



Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Select one:

- Zerind, Oradea, Sibiu, Fagaras, Bucharest
- Zerind, Arad, Sibiu, Rimnicu Vilcea, Pitesti, Bucharest
- Zerind, Oradea, Sibiu, Rimnicu Vilcea, Pitesti, Bucharest
- Zerind, Arad, Sibiu, Fagaras, Bucharest ✔

Your answer is correct. Nice!

Solution:

Greedy best-first search expands nodes according to the heuristic function $h(n)$ = estimated cost to Bucharest (straight-line distances, SLD).

1. Zerind (start state)

2. Arad (SLD(Arad)=366 < SLD(Oradea)=380)

3. Sibiu (SLD(Sibiu)=253 < SLD(Timisoara)=329)

4. Fagaras (SLD(Fagaras)=178 < SLD(Rimnicu Vilcea)=193) < (the other children of Sibiu are backtracking, but our algorithm would check them anyway to be safe))

5. Bucharest

The correct answer is: Zerind, Arad, Sibiu, Fagaras, Bucharest

Question **2**

Correct

Mark 3.00 out of 3.00

Suppose you have three candidate heuristic functions, $h_1(n)$, $h_2(n)$ and $h_3(n)$. All of them are consistent (and therefore admissible/optimistic). You do not know whether or not any of them dominate the others.

Which of the following is the best choice of heuristic function you could use for A* search?

Select one:

- For each node n, take $h(n) = \text{mean}(h_1(n), h_2(n), h_3(n))$
- For each node n, take $h(n) = \min(h_1(n), h_2(n), h_3(n))$
- Pick whichever heuristic has the maximum value for any n.
- For each node n, take $h(n) = \max(h_1(n), h_2(n), h_3(n))$ ✔

Your answer is correct. Nice!

Solution:

The complexity of A* scales with the error, $h^*(n) - h(n)$.

The error of A* scales with how closely the heuristic functions estimate the true costs of solutions from a given node, $h^*(n)$.

Since the heuristics are consistent, they never overestimate the true cost, so $0 \leq h(n) \leq h^*(n)$.

Thus, the **larger** the heuristic function is that we use, the better A* will perform.

The correct answer is: For each node n, take $h(n) = \max(h_1(n), h_2(n), h_3(n))$

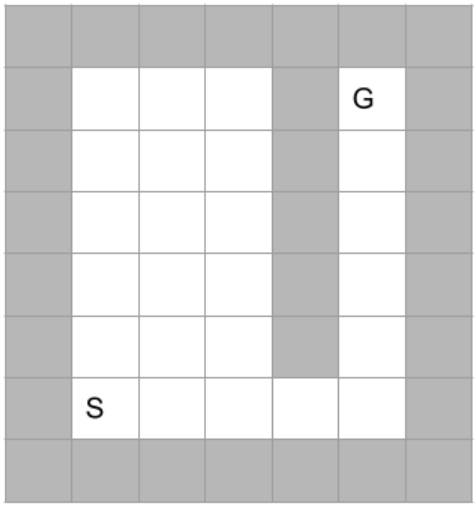
Question **3**

Correct

Mark 3.00 out of 3.00

Which search algorithms that we have learned so far would successfully navigate this maze? S denotes the start state and G denotes the goal state. "Success" here is defined as finding any solution.

Assume that greedy best-first search and A* search both use the straight-line distance to the goal as the heuristic function (we'll talk about some more realistic heuristic functions later).



Select one or more:

- [ ] Greedy best-first search
- [x] Breadth-first search ✔
- [x] Depth-first search ✔
- [x] Uniform-cost search ✔
- [x] A* search ✔

Your answer is correct. Nice!

Solution:

Greedy best-first search is the only one of our algorithms so far that would get stuck in a finite state.

The correct answers are: Breadth-first search, Depth-first search, Uniform-cost search, A* search