Week 2

9 試題

1 point

1 。
The missionaries and cannibals problem is usually stated as follows: 3 missionaries and 3 cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one side outnumbered by the cannibals.

(a) Define a state as a 3 tuple (consist of 2 integers and 1 character), that lists the number of cannibals, missionaries, and the character can be 'r' and 'l', denotes which side the boat is. The goal state is (3, 3, r).

ex: (2, 2, r) means 2 cannibals, 2 missionaries are at the right side and the boat is at the right side.

How many valid and reachable states are there (a search path is stopped once it meets the goal state)?

- 13
- 12
- 15
- 14

1 point

2 。
(b) What is the minimum time for the boat crossed the river to reach the goal?

- 17

- 11
- 13
- 15

1 point

3 。
Does completeness still hold even if zero step costs are allowed in BFS, DFS and the uniform-cost search (UCS)?

- BFS: Yes; DFS: No; UCS: No
- BFS: Yes; DFS: Yes; UCS: Yes
- BFS: Yes; DFS: Yes; UCS: No
- BFS: Yes; DFS: No; UCS: Yes

1 point

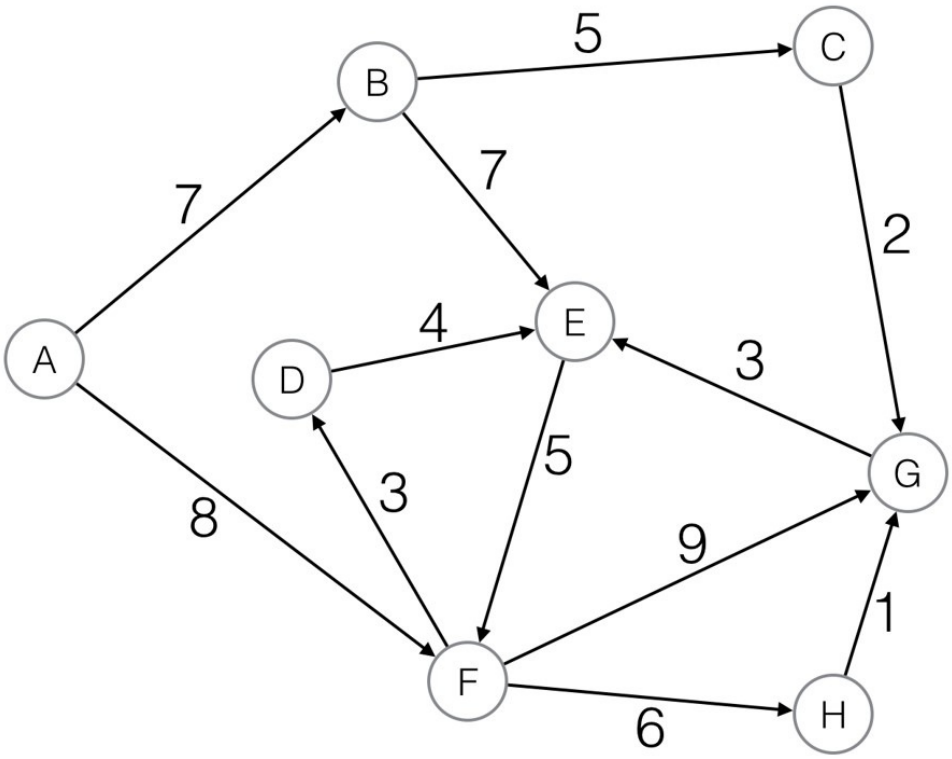
4 。
If the memory is limited, which search is more preferred?

- The depth-Limited search
- BFS
- The bidirectional search
- DFS

1 point

5 。

Use the uniform-cost search to find the minimum path cost from node A to node G



- 15
- 13
- 18
- 14

1 point

6 ◦

Consider the case where step costs are arbitrary (including negative), but no cyclic path exists.

(a) Why doesn't the optimality of BFS, uniform-cost search hold?

- BFS takes too much memory, and the uniform-cost search only chooses the current best path to search
- Because BFS only returns the shortest depth solution, and the uniform-cost search only does the greedy decision.

- The uniform-cost search and BFS can only apply to step cost > 0
- Any path, no matter how bad it appears, might lead to an arbitrarily large reward (negative cost). Therefore, we would need to exhaust all possible paths to ensure to find the optimal path.

---

1 point

7 ◦

(b) Corresponding to the above question,

what can we do to ensure the optimality?

- Simply explore the entire state space
- If we know the range of negative costs, we can add a constant to all step costs so that no step cost is negative.
- Take the first negative step (if any) encountered.
- Run BFS and the uniform-cost search, and then choose the best of the results.

---

1 point

8 ◦

In the original version of IDS, we add the search depth by 1 for each iteration. Now we change this policy by making the depth as the previous depth*2.

(a) What is the completeness and optimality after using this policy?

(**every step cost is 1**)

- Completeness: No; optimality: No
- Completeness: Yes; optimality: No
- Completeness: No; optimality: Yes
- Completeness: Yes; optimality: Yes

1 point

9 。

(b) Corresponding to the above question,

how will this policy change the time and space complexity?

- Time complexity: O(b^d); space complexity: O(bd)
- Time complexity: O($b^{(d-1)}$); space complexity: O(bd)
- Time complexity: O(bd); space complexity: O(bd)
- Time complexity: O(b); space complexity: O(d)

I, **YIJHEN LIN**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account. 了解榮譽準則的更多信息

9 試題 未回答

提交測試