A\* tree search is NOT optimal if...

**A**

The heuristic function is not admissible.

**B**

The heuristic function is not consistent.

**C**

The heuristic function h(n) = 0 for all states.

**D**

All steps have the same cost.

@@@

We're solving the 4-queen problem. An action is defined as putting a queen on an empty square of the board. Considering this initial state, how many successors do we have?

**A**

2

**B**

4

**C**

8

**D**

16

Ảnh có chứa hình vuông, Hình chữ nhật, mẫu, tòa nhà

Mô tả được tạo tự động

@@@@@@@

We're solving the 4-queen problem. An action is defined as putting a queen on an empty square of the board. Which of the following statement is correct regarding the following situation?

**A**

BFS/DFS continues to put the 3rd queen on an empty square of the board.

**B**

Backtracking will declare constraint violation and backtrack to put the 2nd queen on a different square.

**C**

Both are correct.

**D**

None is correct.

Ảnh có chứa hình vuông, Hình chữ nhật, ô gạch, thiết kế

Mô tả được tạo tự động

@@@@

Why is BFS NOT a suitable algorithm for solving N-queen problems?

**A**

BFS is not a complete search algorithm.

**B**

BFS is not an optimal search algorithm.

**C**

Goal nodes of N-queen problems locate at the deepest levels of the search trees.

**D**

None is the correct reason.

@@@@

Why is DFS better than BFS in solving N-queen problems?

**A**

The time complexity of DFS is better than BFS.

**B**

The space complexity of DFS is better than BFS.

**C**

DFS is a complete search algorithm.

**D**

DFS is an optimal search algorithm.

@@@@

What is the main drawback of the normal DFS algorithm in solving N-queen problems?

**A**

DFS might loop forever on the search tree.

**B**

All goal states locate at the deepest level of the search tree.

**C**

Constraint violations are not checked during the search.

**D**

All of the above are correct.

@@@@

We're solving the 4-queen problem. An action is defined as putting a queen on square in the leftmost empty column of the board such that no queen can attack one another. How many successors do we have regarding the current board?

**A**

1

**B**

2

**C**

3

**D**

4

Ảnh có chứa hình vuông, Hình chữ nhật, mẫu, ô gạch

Mô tả được tạo tự động

@@@@

We're solving the 4-queen problem. An action is defined as putting a queen on a square in the left most empty column of the board such that no queen can attack one another. All the squares in the 2nd column have been tried out. Regarding the current board, what do you think the backtracking algorithm would do next?

**A**

Putting the 4th queen on a square in the 4th column.

**B**

Backtracking to one level higher up the search tree, and then putting the 3rd queen on a different square in the 3rd column.

**C**

Backtracking to two levels higher up the search tree, and then putting the 2nd queen on a different square in the 2nd column.

**D**

Backtracking to three levels higher up the search tree, and then putting the 1st queen on a different square in the 1st column.

