SUBJECT CODE: VBDS1402

MODULE 1

CHAPTER 2

SOLVING PROBLEMS BY SEARCHING

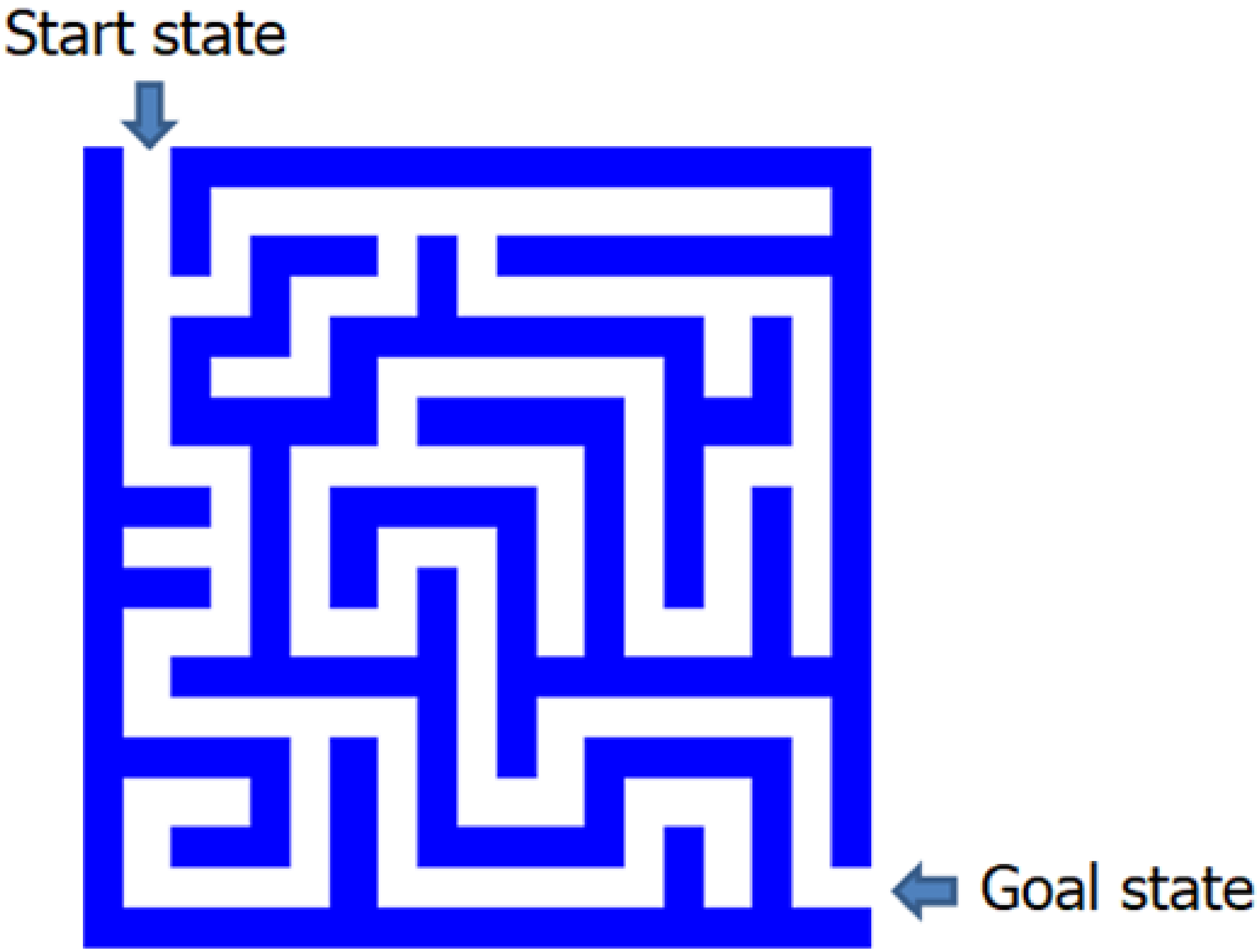
ARTIFICIAL INTELLIGENCE

IN THIS LECTURE, YOU WILL LEARN:

- Various terms like problem formulation, search, solution, etc.
- Components of a problem
- Some more terms like abstraction, real world problems
- How problems are formulated?
- How toy problems are formulated?
- How 8 puzzle problem is formulated?

What is the first step in problem solving?	• Goal formulation, based on the current situation and the agent's performance measure, is the first step in problem solving.
Define problem formulation	• Process of deciding what actions and states to consider, given a goal.
Define search	• The process of looking for a sequence of actions that reaches the goal.
Define solution	• A solution is an action sequence, so search algorithms work by considering various possible action sequences.
Define execution phase	• Once a solution is found, the actions it recommends can be carried out. This is called the execution phase.

SEARCH



COMPONENTS OF A PROBLEM

Initial State The initial state that the agent starts in.

Actions A description of the possible actions available to the agent.

Transition model A description of what each action does; the formal name for this is the transition model, specified by a function RESULT('*s*', '*a*') that returns the state that results from doing action '*a*' in state '*s*'.

Goal test The goal test, which determines whether a given state is a goal state.

Path Cost A path cost function that assigns a numeric cost to each path.

SOME MORE TERMS

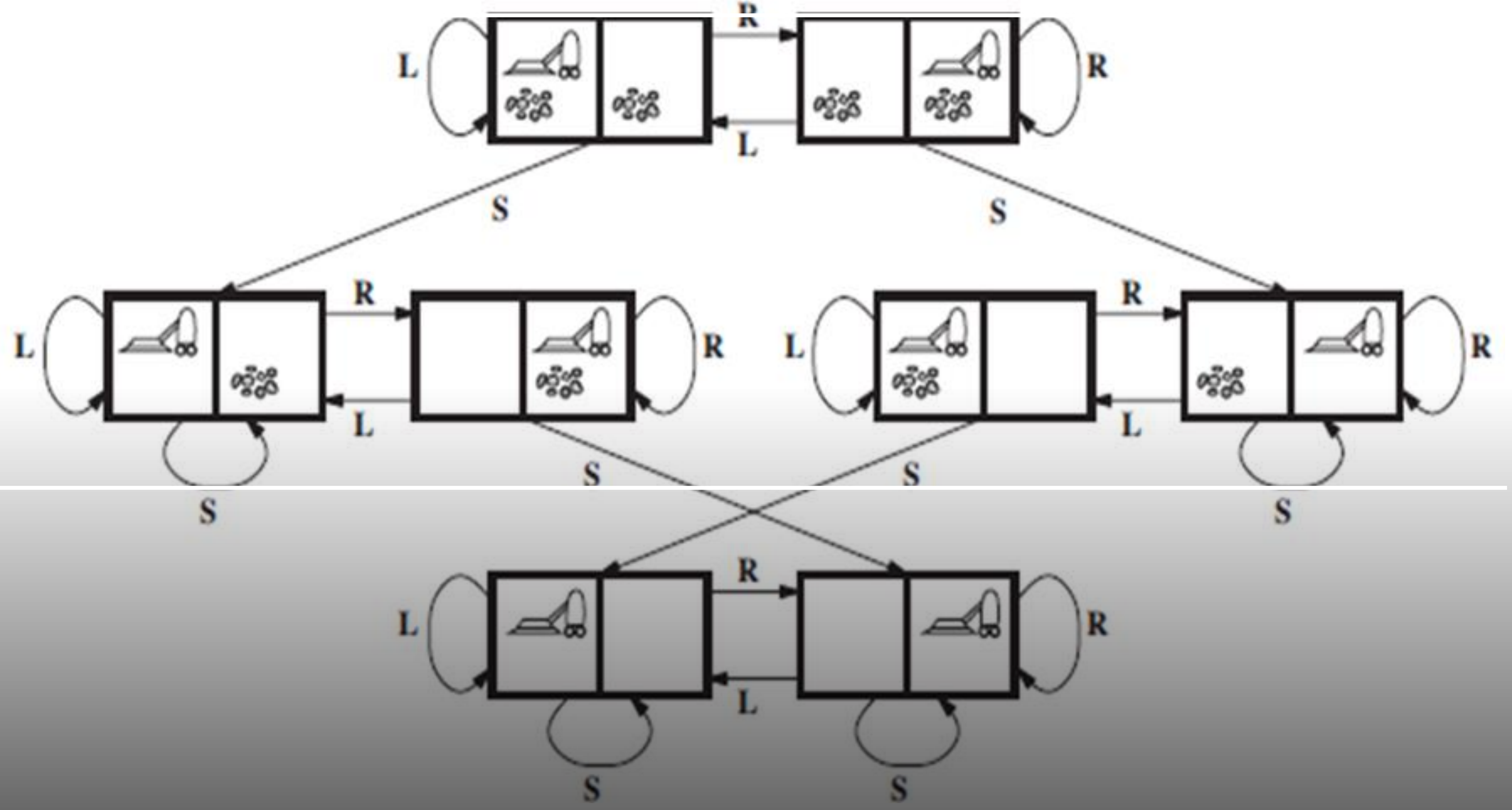
Abstraction

The process of removing detail from a representation

Real world problem

A real-world problem is one whose solutions people actually care about

HOW ARE PROBLEMS FORMULATED?



HOW TOY PROBLEMS ARE FORMULATED?

HOW TOY PROBLEMS ARE FORMULATED?

States: The state is determined by both the agent location and the dirt locations. The agent is in one of two locations, each of which might or might not contain dirt.

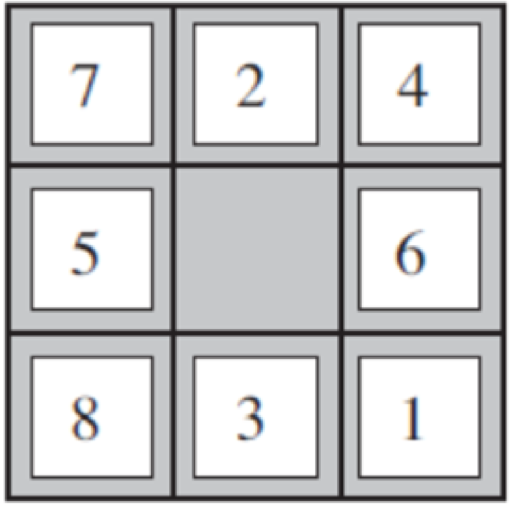
Initial state: Any state can be designated as the initial state.

Actions: In this simple environment, each state has just three actions: Left, Right, and Suck. Larger environments might also include Up and Down.

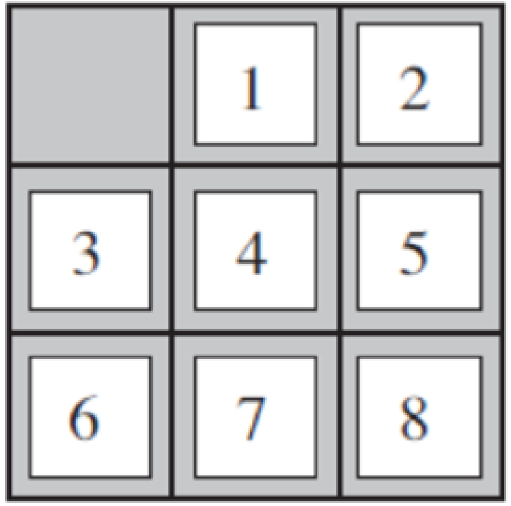
Transition model: The actions have their expected effects, except that moving Left in the leftmost square, moving Right in the rightmost square, and Sucking in a clean square have no effect.

Goal test: This check whether all the squares are clean.

Path cost: Each step costs 1, so the path cost is the number of steps in the path



Start State



Goal State

HOW IS 8 PUZZLE PROBLEM FORMULATED?

HOW IS 8 PUZZLE PROBLEM FORMULATED?

- **States:** A state description specifies the location of each of the eight tiles and the blank in one of the nine squares.
- **Initial state:** Any state can be designated as the initial state. Note that any given goal can be reached from exactly half of the possible initial states.
- **Actions:** The simplest formulation defines the actions as movements of the blank space Left, Right, Up, or Down. Different subsets of these are possible depending on where the blank is.
- **Transition model:** Given a state and action, this returns the resulting state; for example, if we apply Left to the start state in above Figure, the resulting state has the 5 and the blank switched.
- **Goal test:** This check whether the state matches the goal configuration shown in Figure. (Other goal configurations are possible.)
- **Path cost:** Each step costs 1, so the path cost is the number of steps in the path.

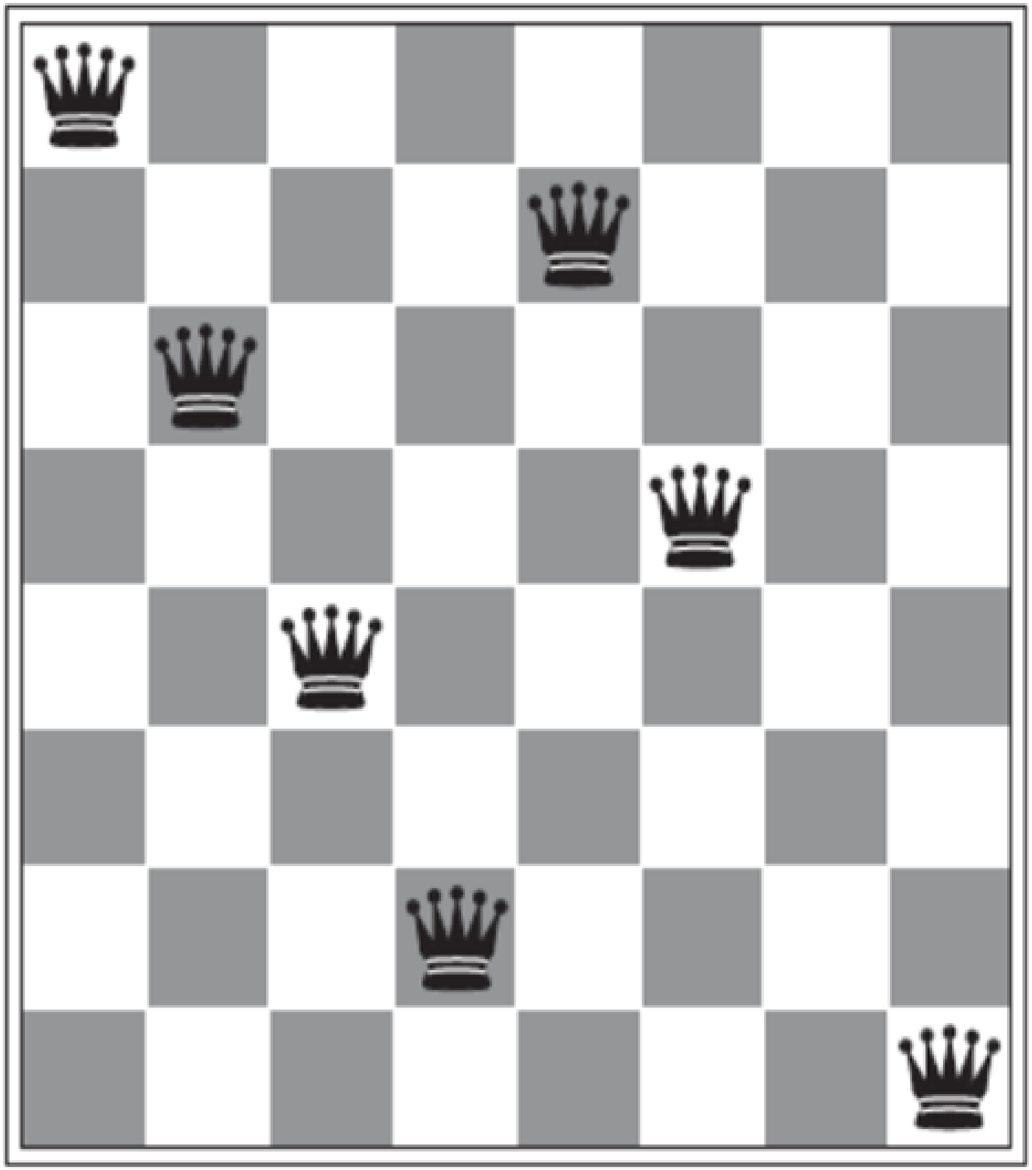
How is 8 - queens problem formulated?

There are two main kinds of formulation:

■An incremental formulation involves operators that augment the state description, starting with an empty state; for the 8-queens problem, this means that each action adds a queen to the state.

■A complete-state formulation starts with all 8 queens on the board and moves them around.

In either case, the path cost is of no interest because only the final state counts.



How is 8 – queens problem formulated?

The first incremental formulation one might try is the following:

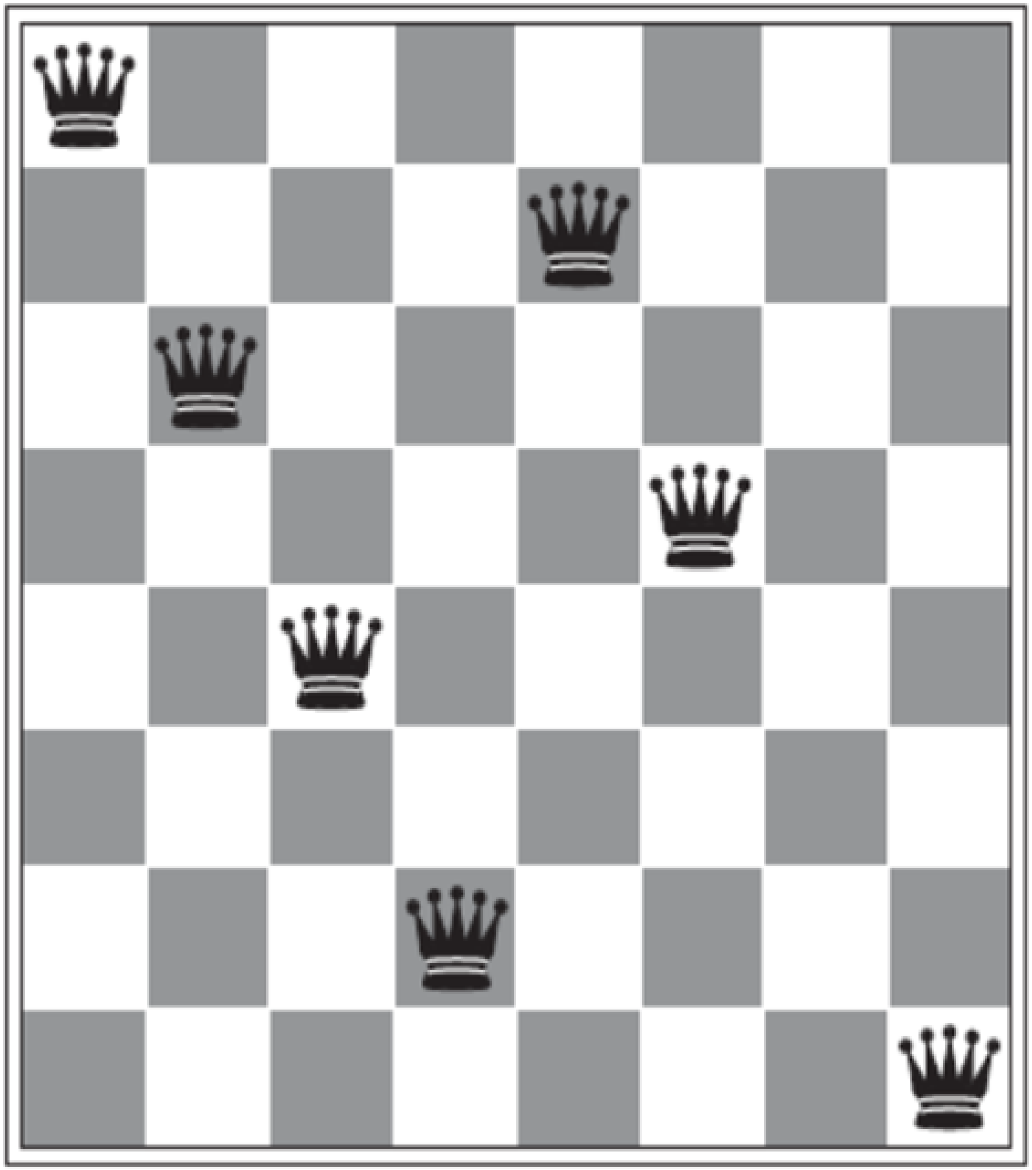
■States: Any arrangement of 0 to 8 queens on the board is a state.

■Initial state: No queens on the board.

■Actions: Add a queen to any empty square.

■Transition model: Returns the board with a queen added to the specified square.

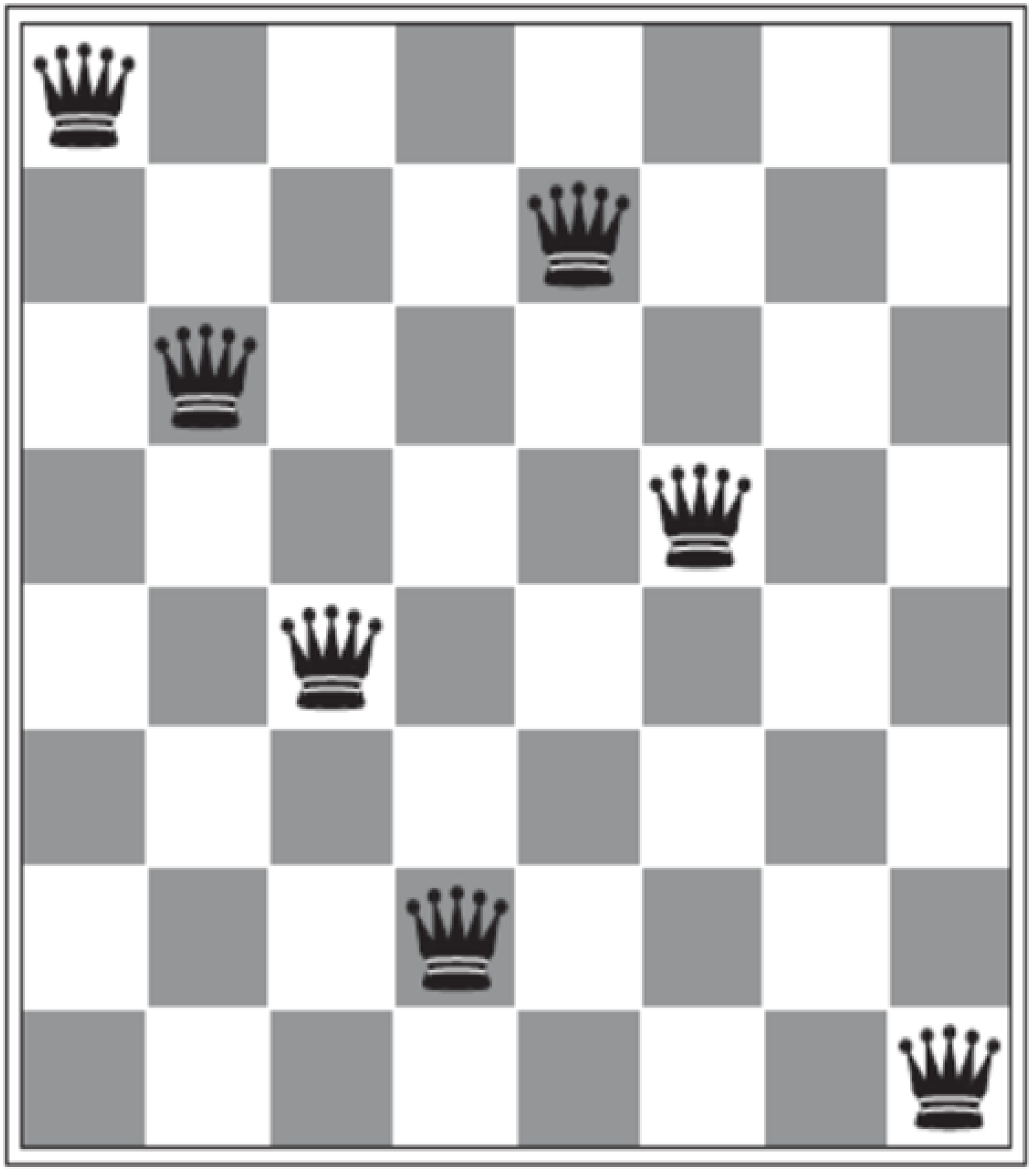
■Goal test: 8 queens are on the board, none attacked.



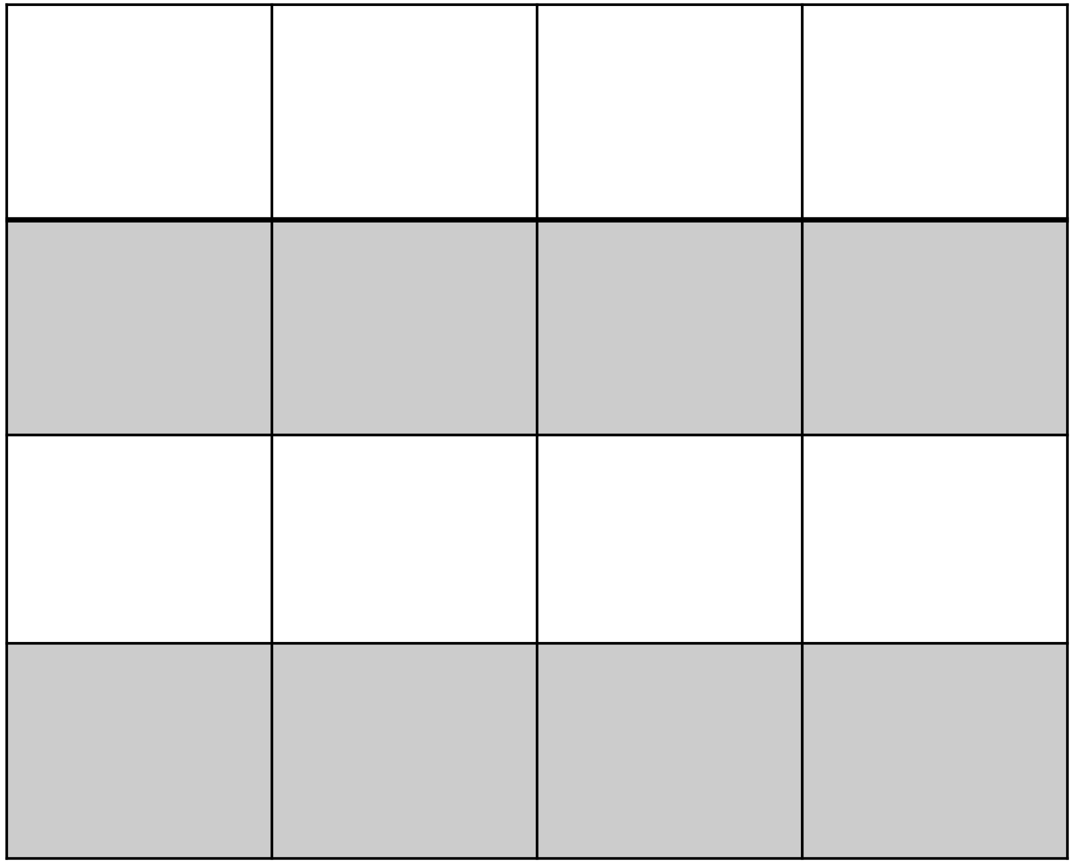
How is 8 - queens problem formulated?

A better formulation would prohibit placing a queen in any square that is already attacked:

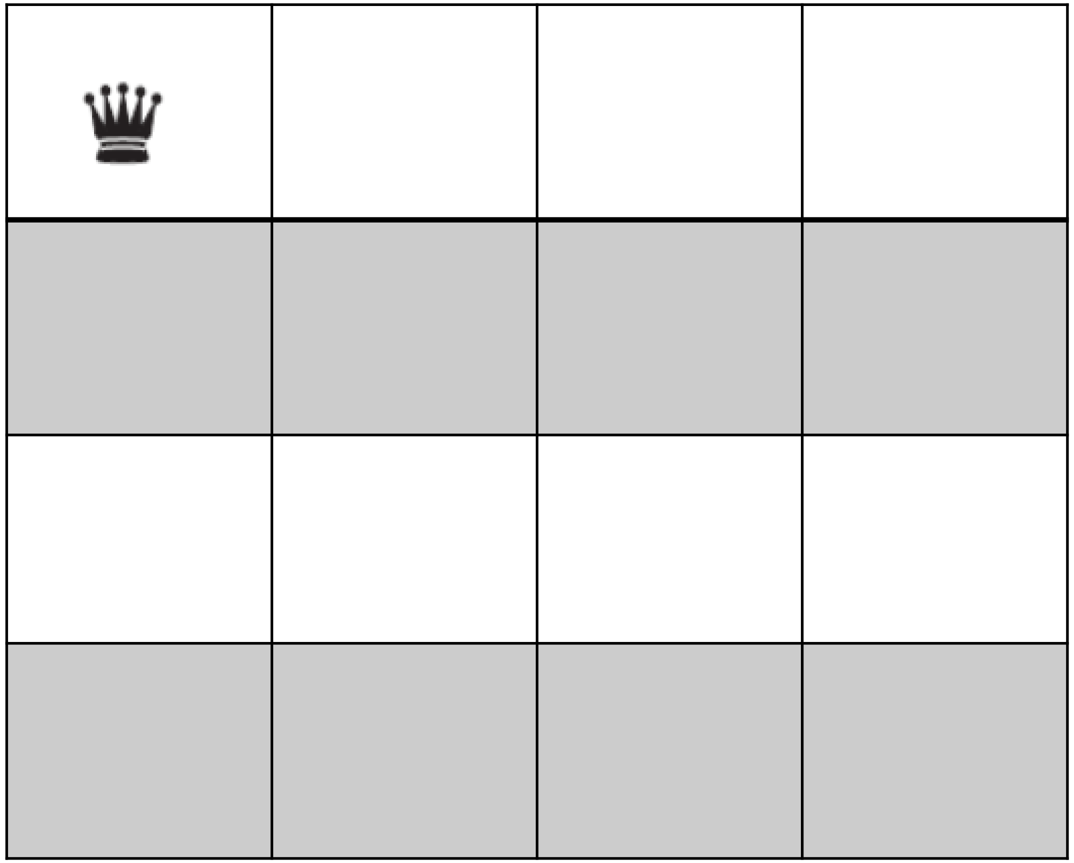
- States: All possible arrangements of n queens (0 = n = 8), one per column in the leftmost n columns, with no queen attacking another.
- Actions: Add a queen to any square in the leftmost empty column such that it is not attacked by any other queen.



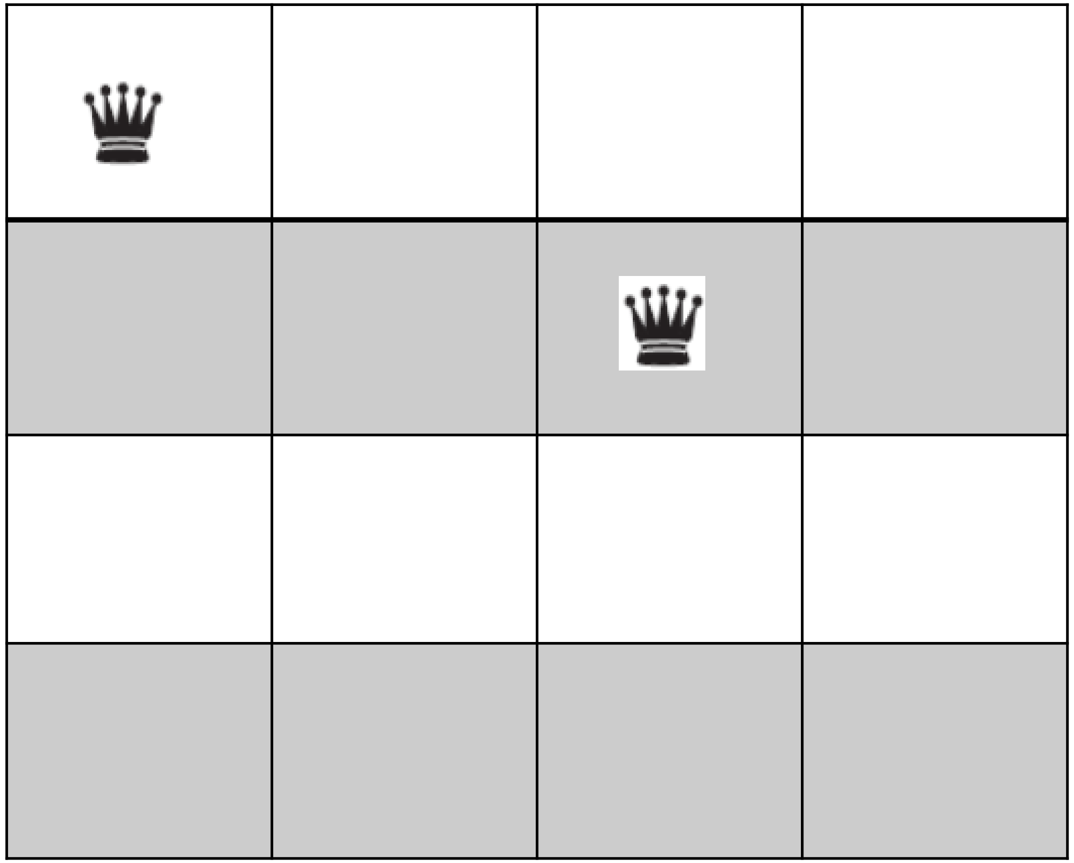
Example of 4 queens



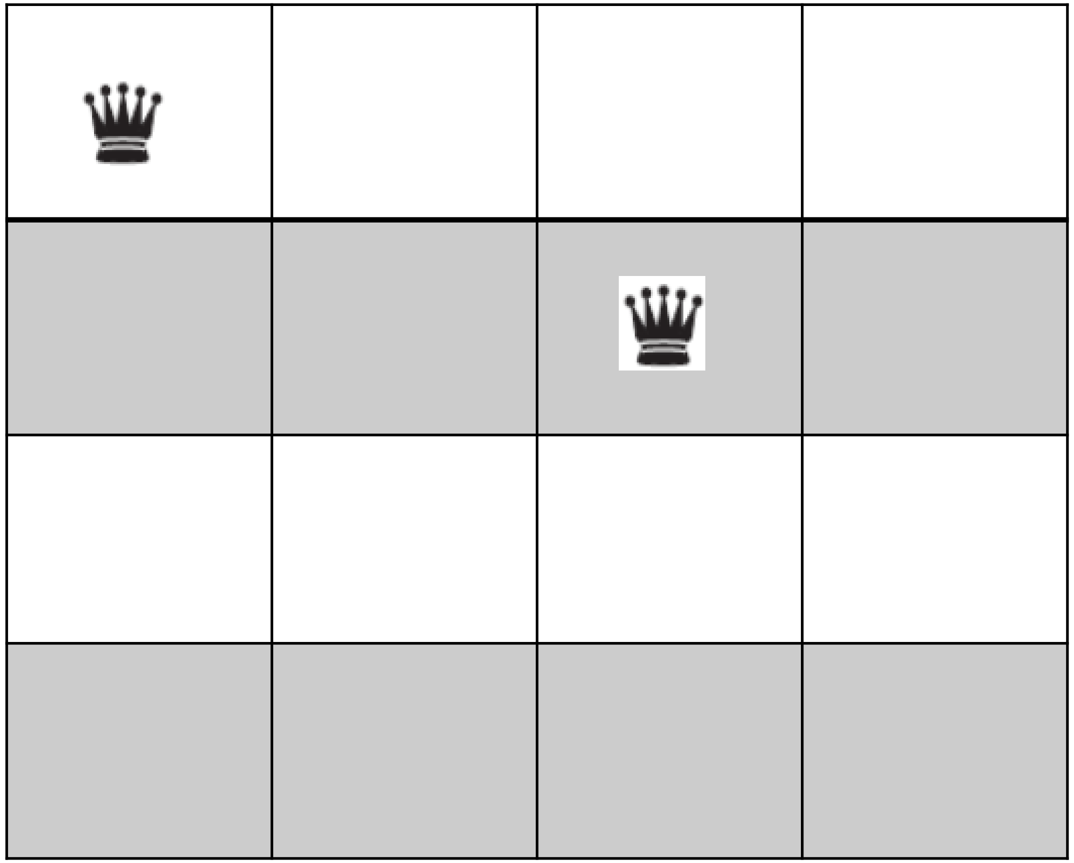
Place the first queen in top left



Place the 2nd queen

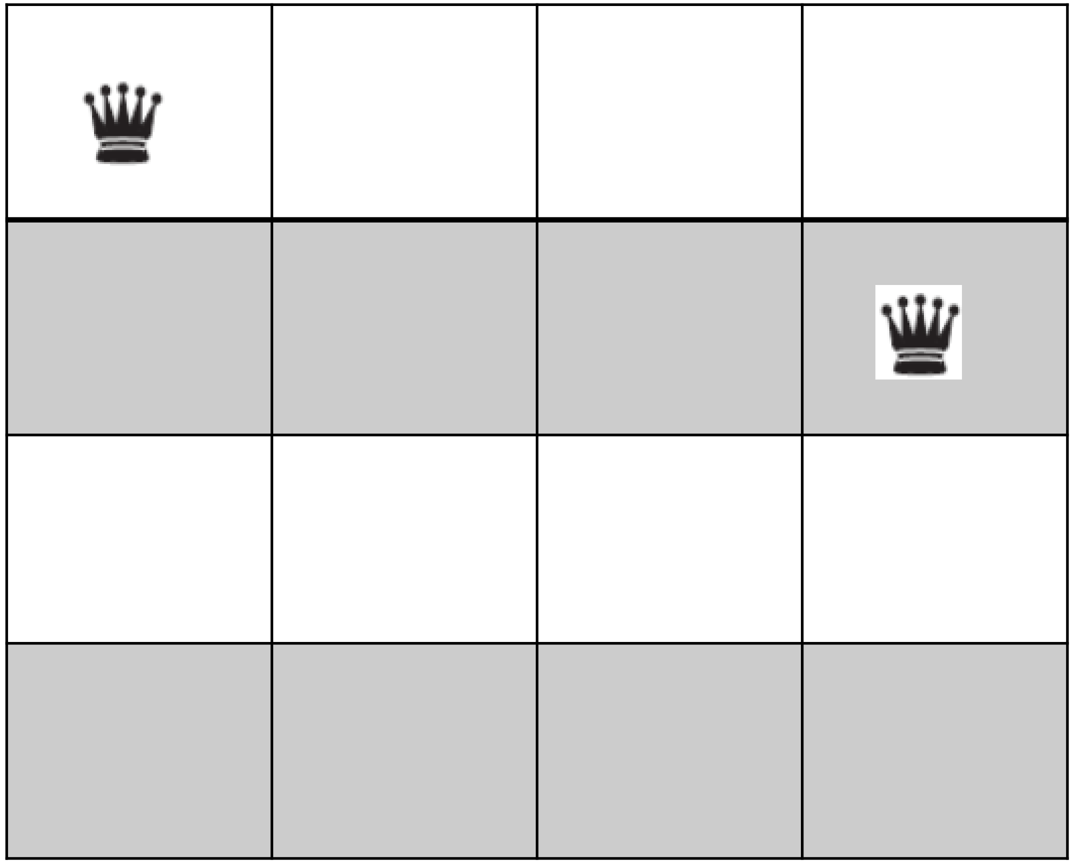


No place for the 3rd queen

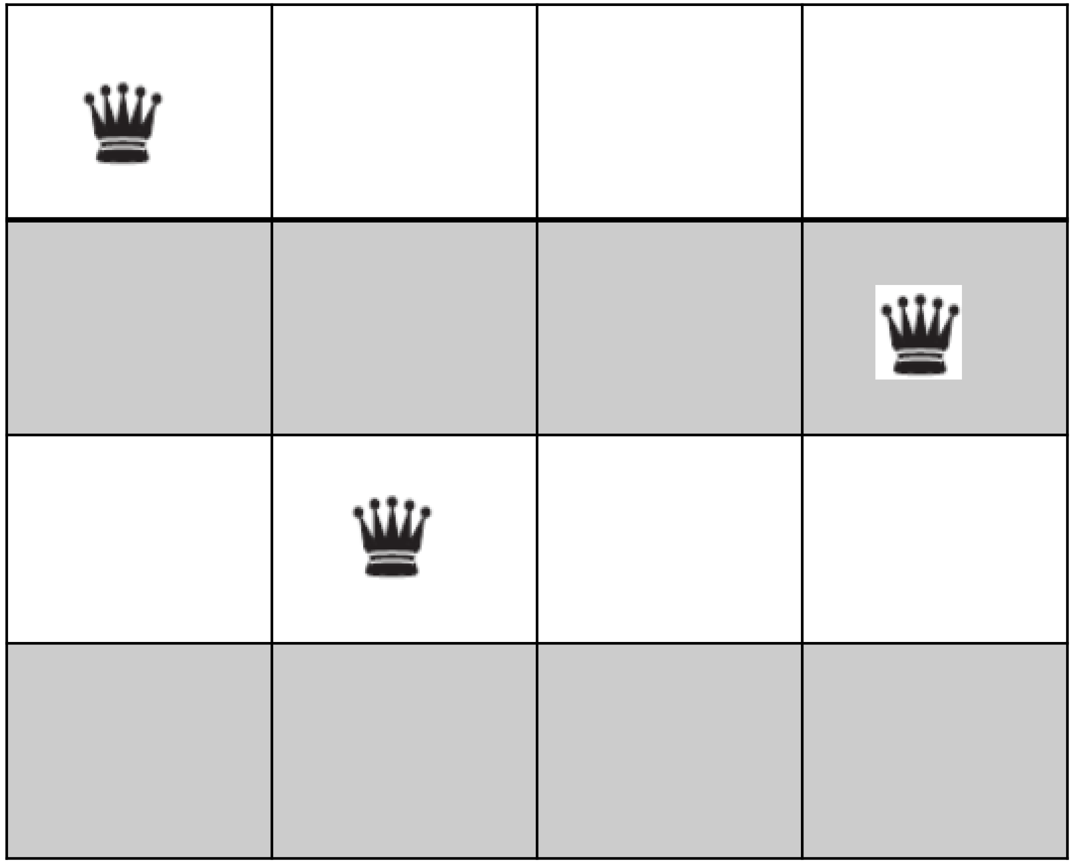


Back Track

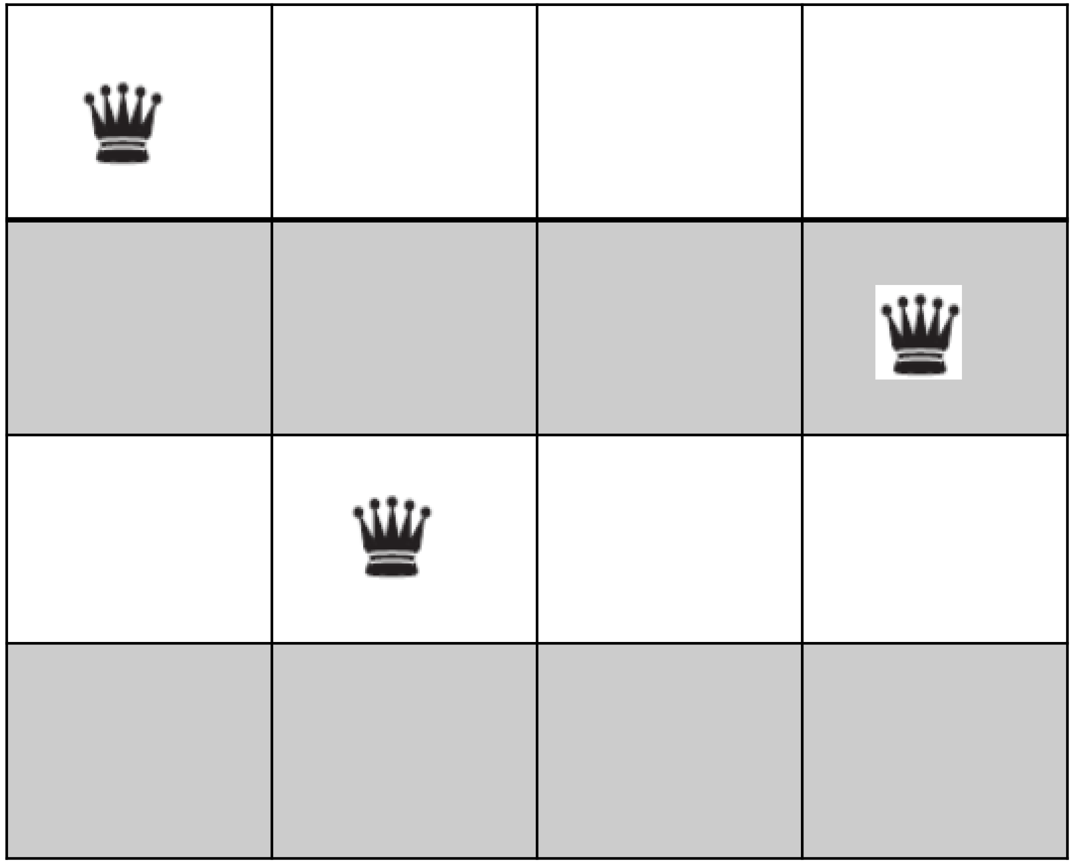
Placed the 2nd queen in the 2nd row 4th col



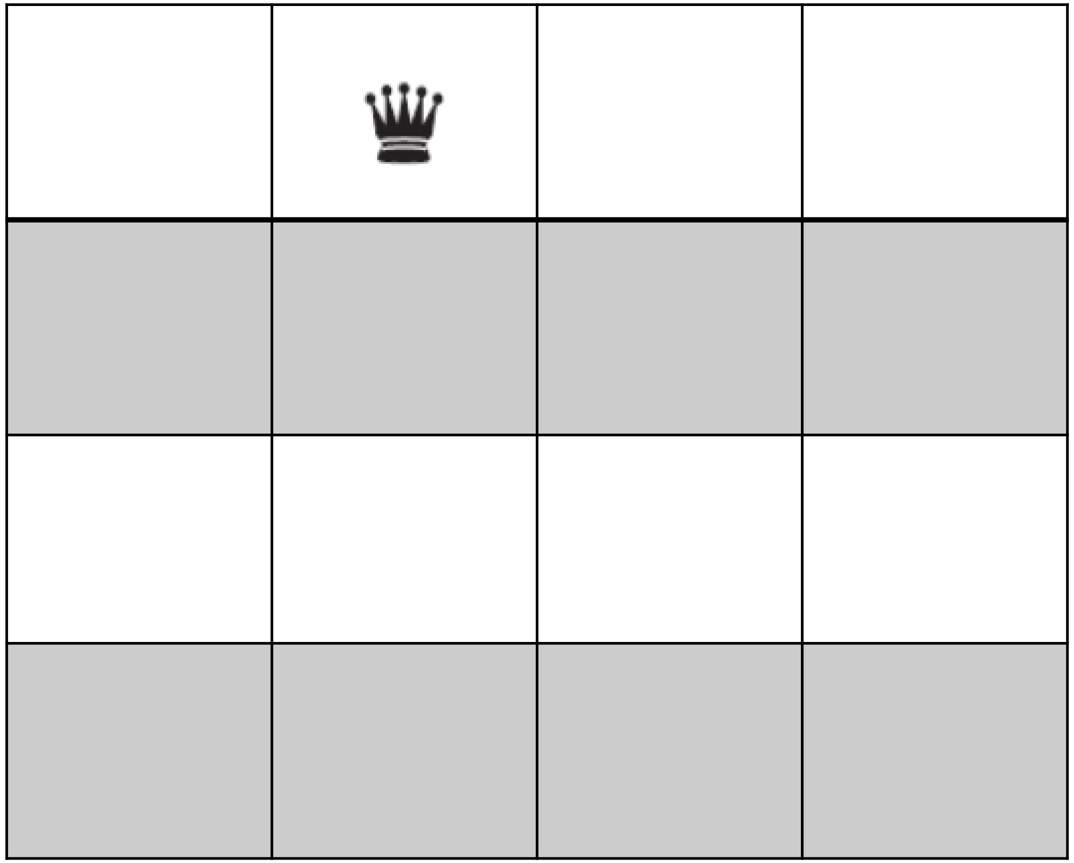
Place the 3rd queen in 3rd row 2nd col



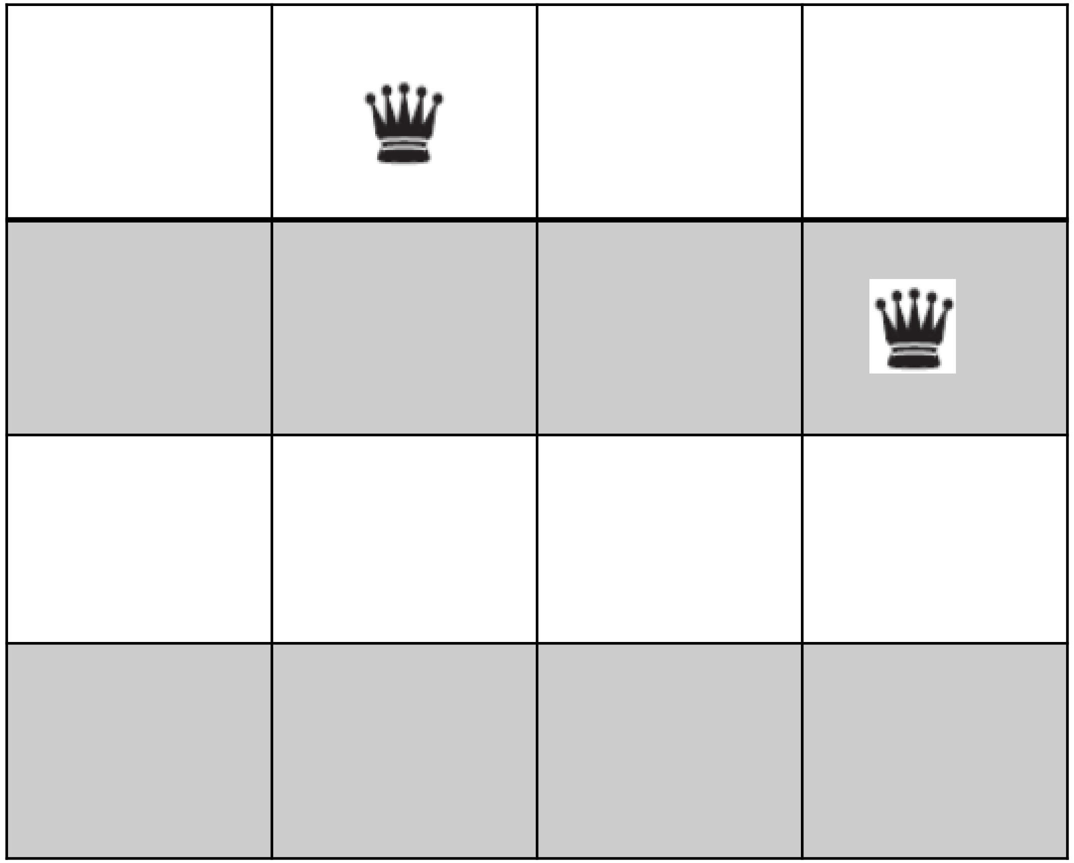
No place for 4th queen



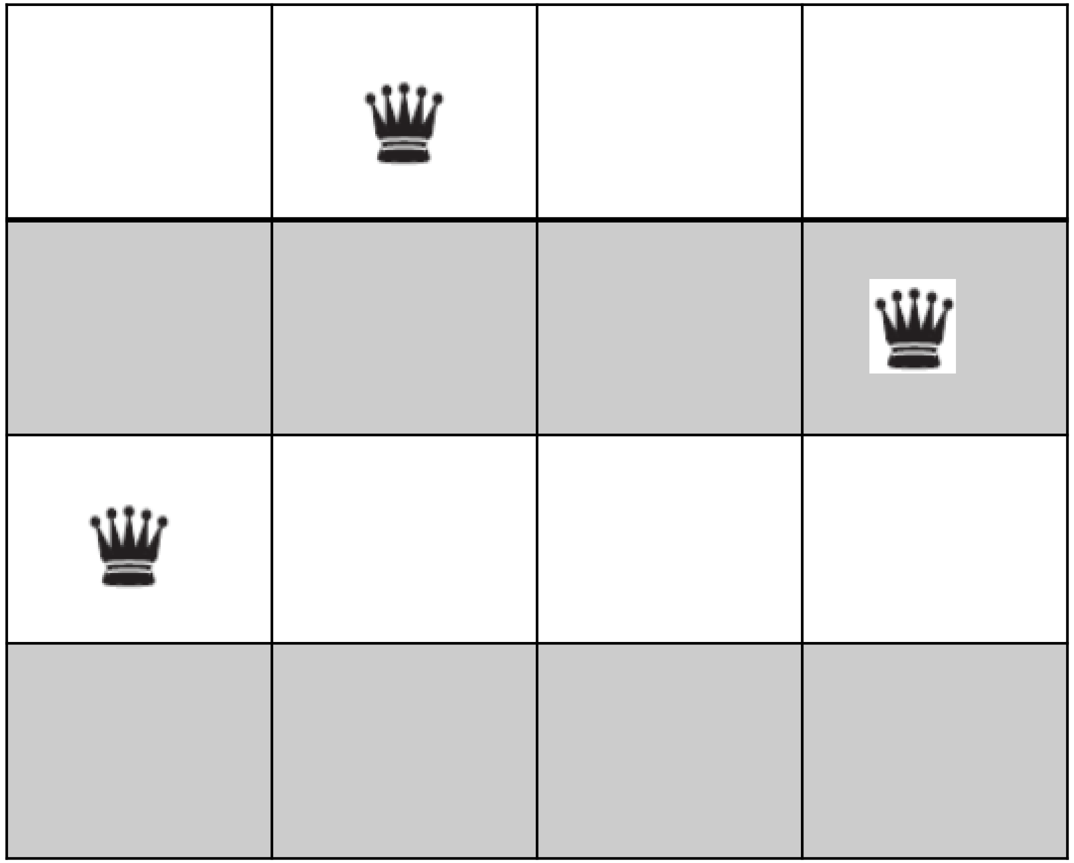
Back track (It means that 1st queen was placed at the wrong place)



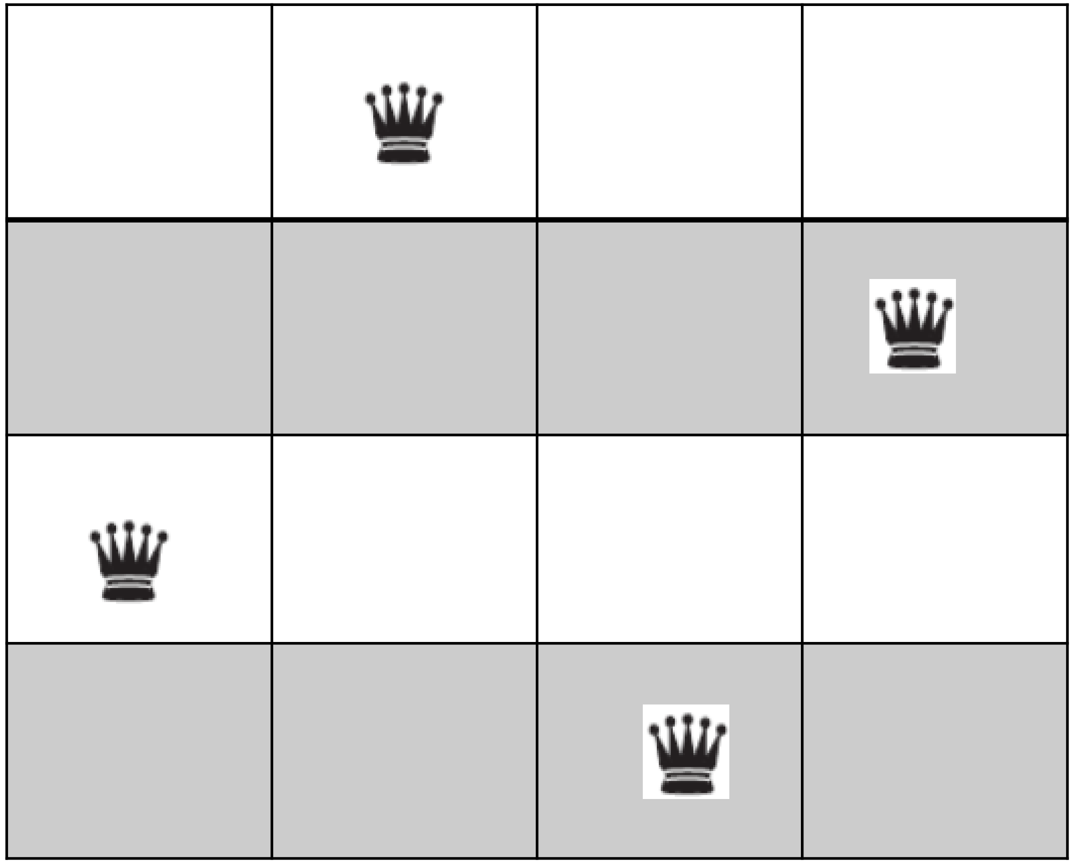
Place the 2nd queen in the 2nd row, 4th col



Place the 3rd queen in the 3rd row, 1st col



Place the 4th queen in the 4th row, 3rd col (A possible solution of 4 queen problem)



CAN YOU FIND ANOTHER SOLUTION FOR 4 QUEEN’S PROBLEM?

How we can reach 5 from 4?

$$\left\lfloor \sqrt{\sqrt{\sqrt{\sqrt{\sqrt{(4!)!}}}}} \right\rfloor = 5\,.$$

The problem definition is very simple:

- States: Positive numbers.
- Initial state: 4.
- Actions: Apply factorial, square root, or floor operation (factorial for integers only).
- Transition model: As given by the mathematical definitions of the operations.
- Goal test: State is the desired positive integer.

For example,

The number 620,448,401,733,239,439,360,000 is generated in the expression for 5—so the state space for this problem is infinite.

Such state spaces arise frequently in tasks involving the generation of mathematical expressions, circuits, proofs, programs, and other recursively defined objects.


THANK YOU