ARTIFICIAL INTELLIGENCE

PROBLEM SOLVING BY SEARCHING

CS-632

University Institute of Information Technology

PMAS-Arid Agriculture University Rawalpindi

PROBLEM SOLVING AGENT

- A kind of Goal-based agent.
- Decide what to do by searching sequences of actions that lead to desirable states.

PROBLEM DEFINITION

- Initial state : starting point
- Operator: description of an action
- State space: set of all states reachable from the initial state by any sequence of actions.
- Path: sequence of actions leading from one state to another
- Goal test: which the agent can apply to a single state description to determine if it is a goal state
- Path cost function: assign a cost to a path which the sum of the costs of the individual actions along the path.

WHAT IS SEARCH?

- Search is the systematic examination of states to find path from the start/root state to the goal state.
- The set of possible states, together with *operators* defining their connectivity in the *search space*.
- The output of a search algorithm is a solution, that is, a path from the initial state to a state that satisfies the goal test.
- In real life search usually results from a lack of knowledge. In AI too search is merely an offensive instrument with which to attack problems that we can't seem to solve any better way.

GOAL-FORMULATION

- What is the goal state?
- What are important characteristics of the goal state?
- How does the agent know that it has reached the goal?
- Are there several possible goal states?
  - Are they equal or are some more preferable?
- Courtesy: Dr. Franz J. Kurfess

GOAL

- We will consider a goal to be a set of world states – just those states in which the goal is satisfied.
- Actions can be viewed as causing transitions between world states.

LOOKING FOR PARKING

- Going home; need to find street parking
- Formulate Goal:

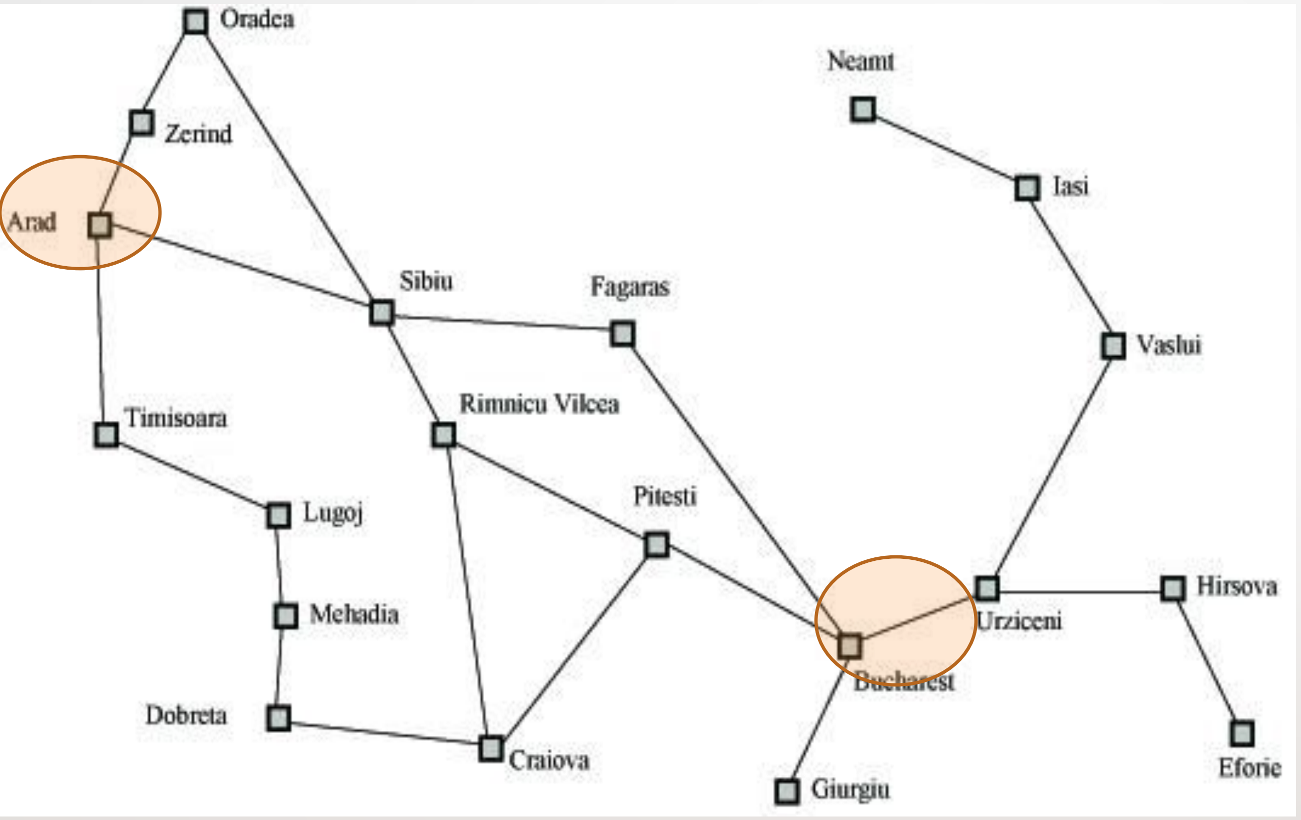
  Car is parked
- Formulate Problem:

  States: street with parking and car at that street

  Actions: drive between street segments
- Find solution:

  Sequence of street segments, ending with a street with parking

SEARCH EXAMPLE



Formulate goal: Be in Bucharest.

Formulate problem: states are cities, operators drive between pairs of cities

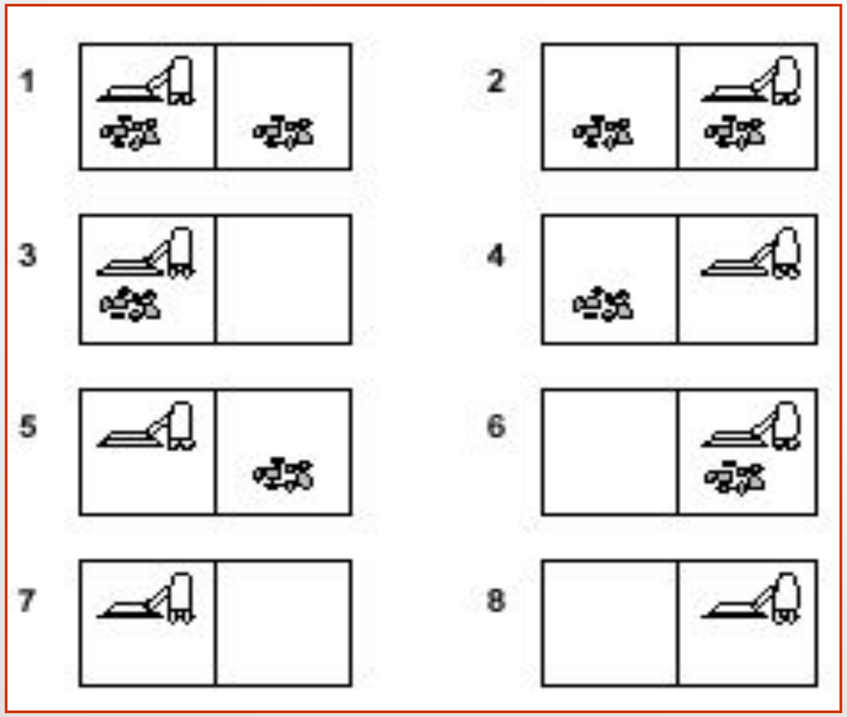
Find solution: Find a sequence of cities (e.g., Arad, Sibiu, Fagaras, Bucharest) that leads from the current state to a state meeting the goal condition

PROBLEM FORMULATION

A search problem is defined by the

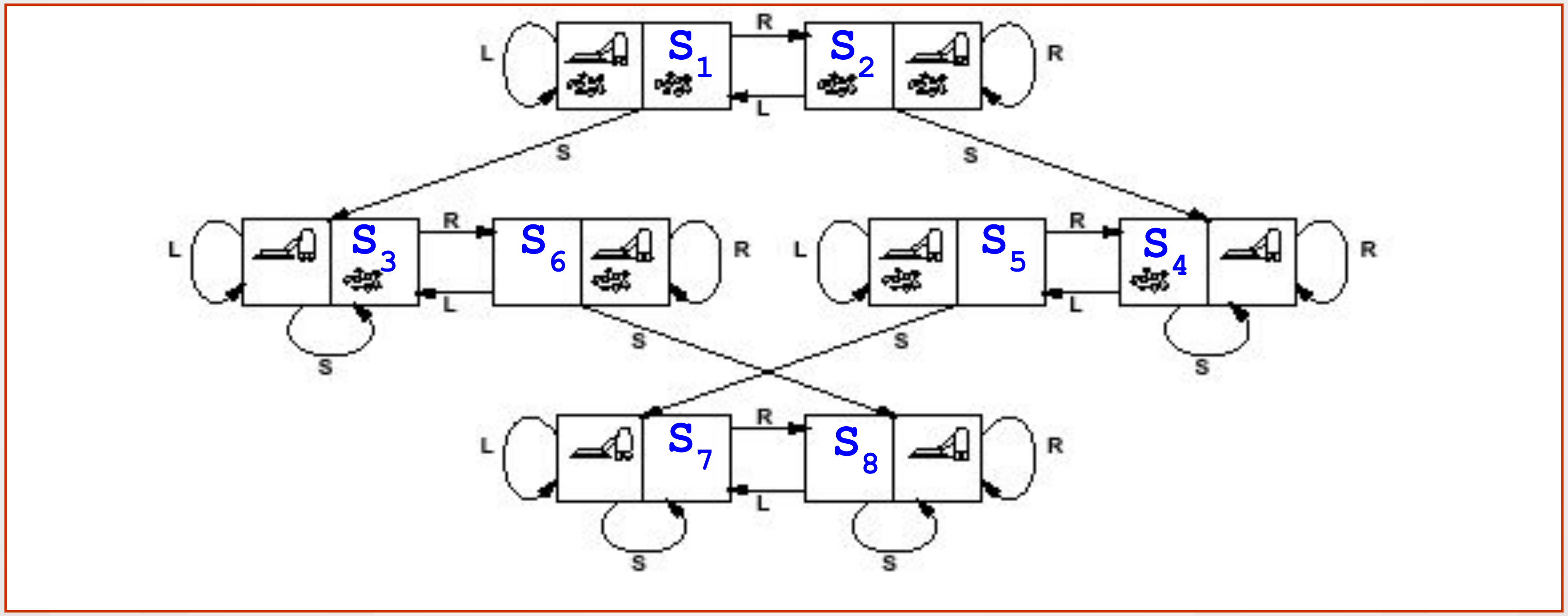
1. Initial state (e.g., Arad)
2. Operators (e.g., Arad -> Zerind, Arad -> Sibiu, etc.)
3. Goal test (e.g., at Bucharest)
4. Solution cost (e.g., path cost)

VACUUM WORLD



- 8 possible world states
- 3 possible actions: *Left/Right/ Suck*
- Goal: clean up all the dirt= state(7) or state(8)

VACUUM WORLD



•**States:** *S_1, S_2, S_3, S_4, S_5, S_6, S_7 , S_8*

• **Operators:** Go Left , Go Right , Suck

• **Goal test**: no dirt left in both squares

• **Path Cost:** each action costs 1.

Problem Spaces

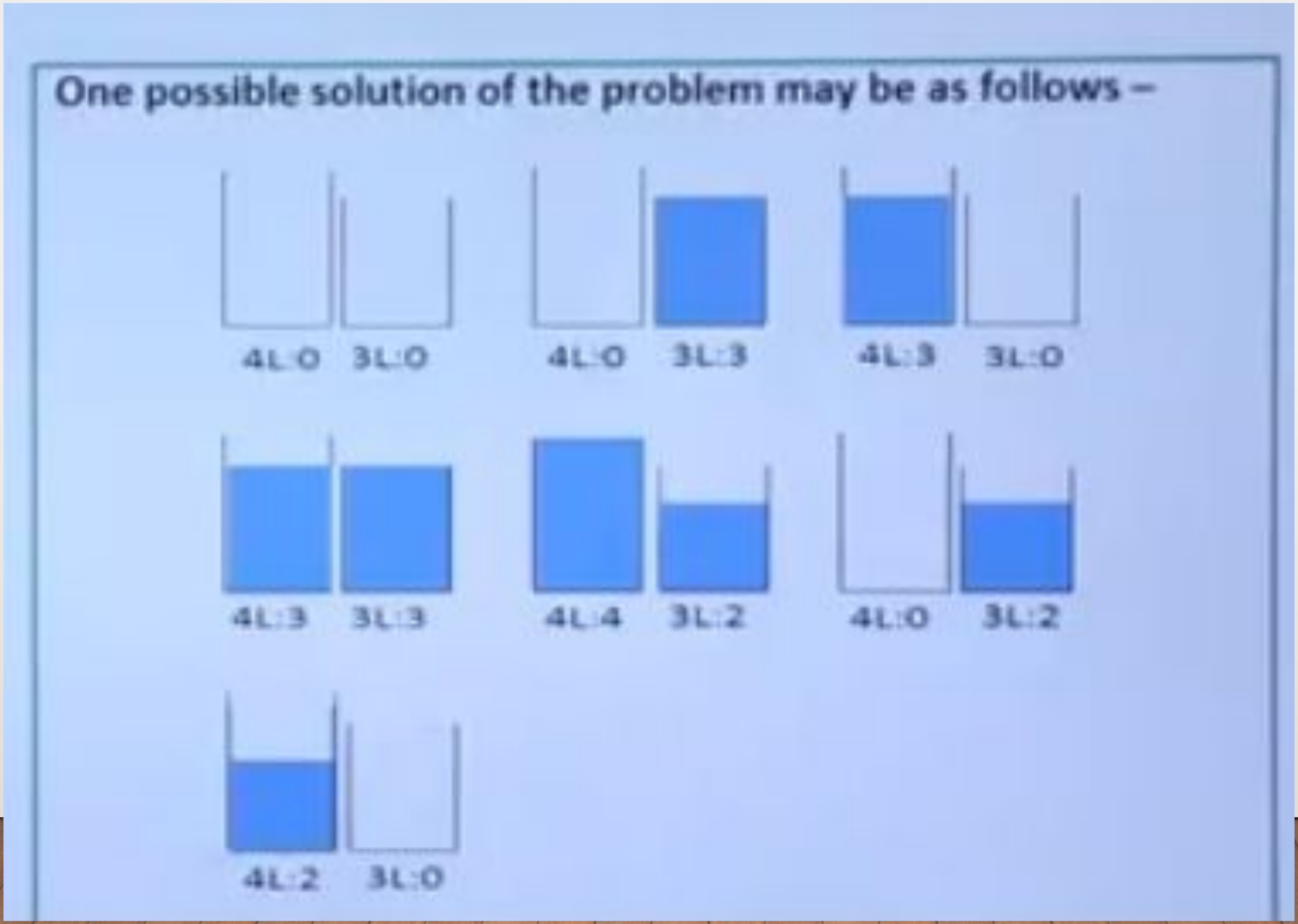
- Water Jug Problem
  - We have two jugs of 4 gallon and 3 gallons capacity
  - The jugs do not have any measuring markers
  - How can we get exactly 2 gallons of water into the 4 gallon jug?

- We can define the possible states for the problem and make a list of legal or valid operations

- The possible states for this problem can be described as the set of ordered pairs of integers (x, y), such that x = 0, 1, 2, 3, or 4 and y = 0, 1, 2, or 3. Here x and y represent the gallons of water in the two jugs

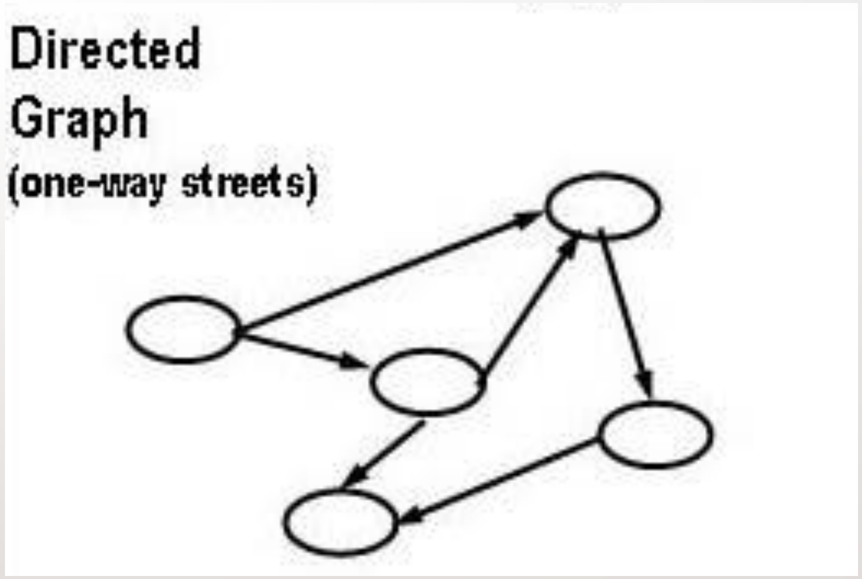
- Our start state is (0,0) and our goal state is (2,n)

WATER JUG PROBLEM


One possible solution of the problem may be as follows –

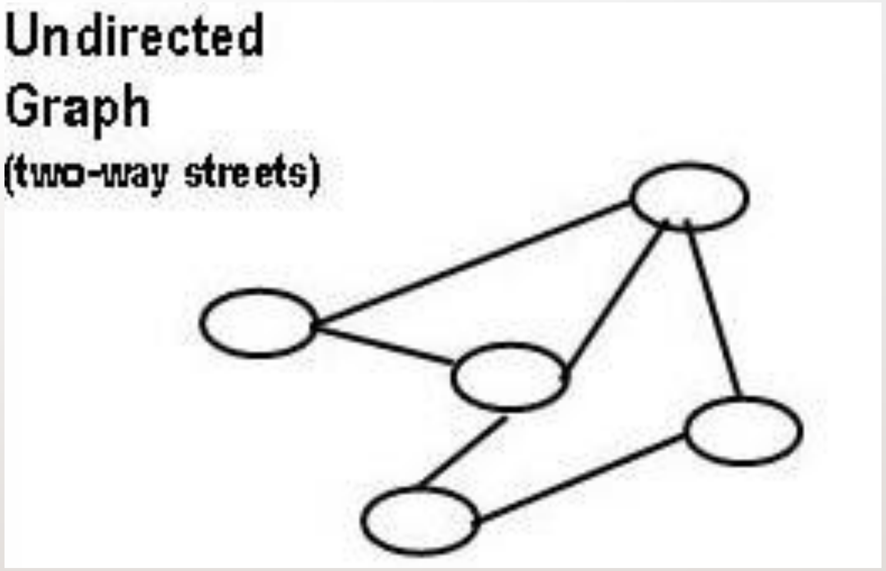
DIRECTED GRAPHS

- A graph is also a set of nodes connected by links but where loops are allowed, and a node can have multiple parents.
- We have two kinds of graphs to deal with: **directed** graphs, where the links have direction (one-way streets).



UNDIRECTED GRAPHS

Undirected graphs where the links go both ways. You can think of an undirected graph as shorthand for a graph with directed links going each way between connected nodes.



GOAL DIRECTED AGENT

- A goal directed agent needs to achieve certain goals.
- Such an agent selects its actions based on the goal it has.
- Many problems can be represented as
  - a set of states and
  - a set of rules of how one state is transformed to another.

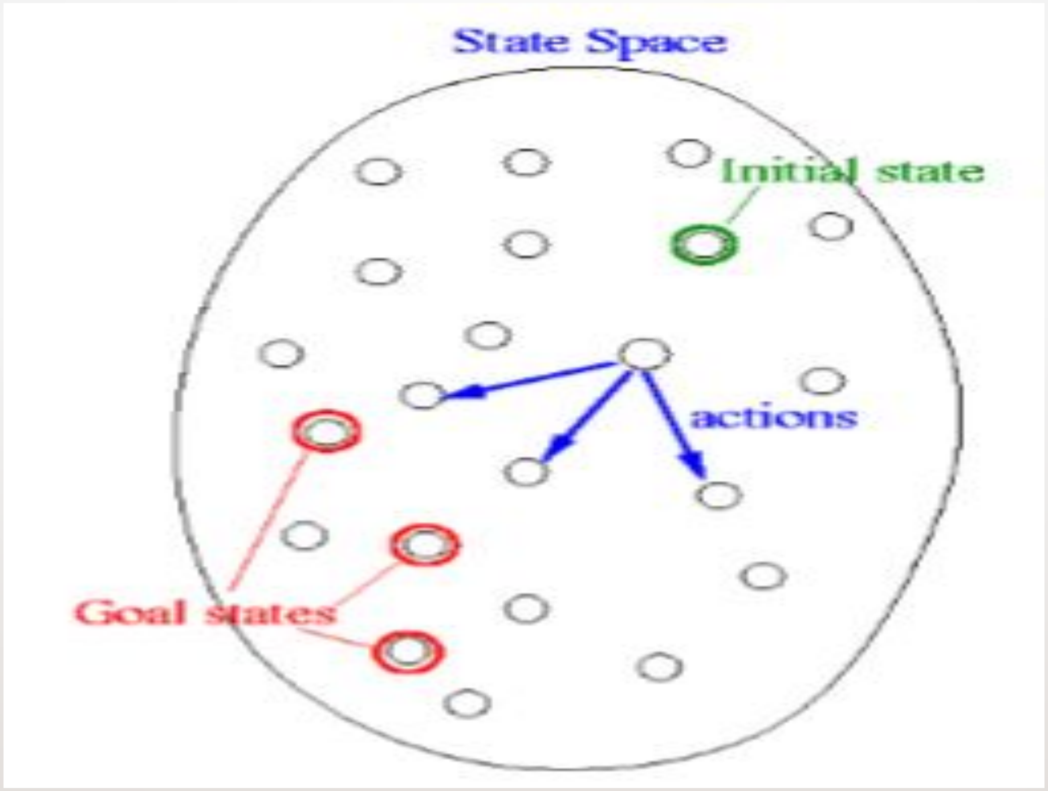
SEARCH PROBLEM

- A search problem consists of the following:
  - S: the full set of states
  - S_0 : the initial state
  - $A:S \rightarrow S$ is a set of operators
  - G is the set of final states. Note that $G \subseteq S$

SEARCH PROBLEM

- The search problem is to find a sequence of actions which transforms the agent from the initial state to a goal state $g \in G$

STATE SPACE REPRESENTATION



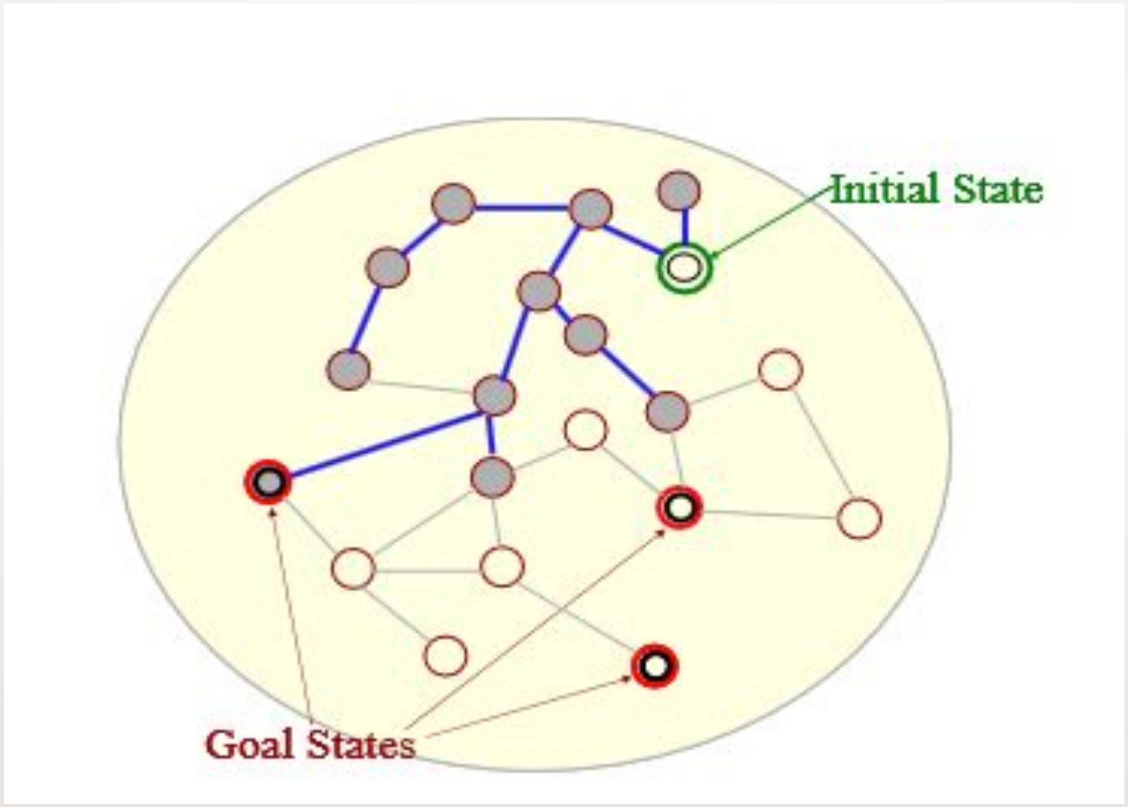
REPRESENTATION OF SEARCH PROBLEMS

- A search problem is represented using a directed graph.
- The states are represented as nodes.
- The allowed actions are represented as arcs.

SEARCHING PROCESS

- The generic searching process can be very simply described in terms of the following steps:
- Do until a solution is found or the state space is exhausted.
    - Check the current state
    - Execute allowable actions to find the successor states.
    - Pick one of the new states.
    - Check if the new state is a solution state. If it is not, the new state becomes the current state, and the process is repeated.

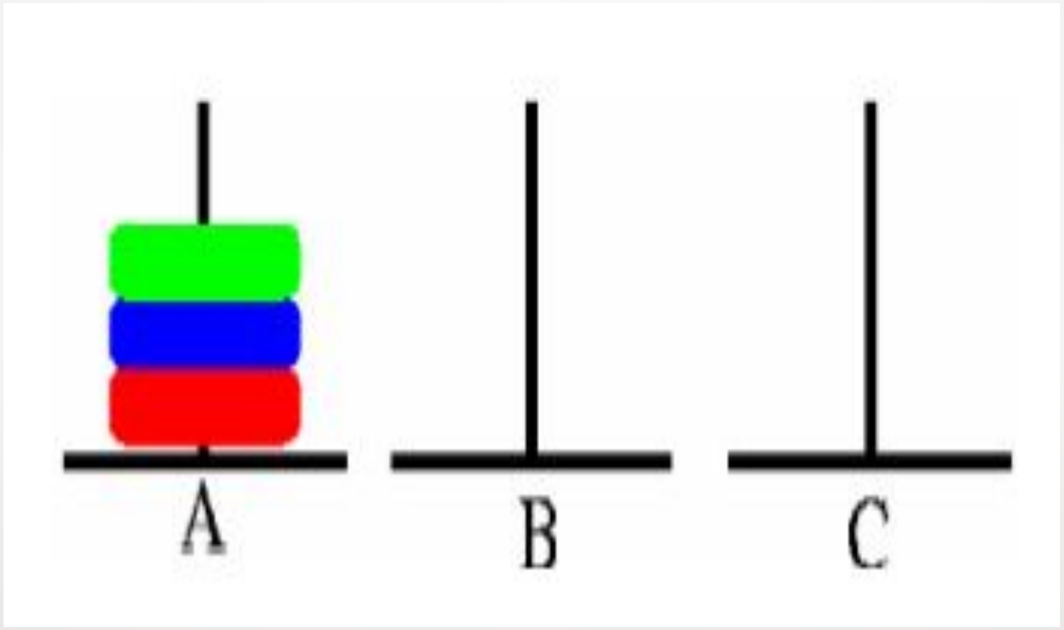
ILLUSTRATION OF A SEARCH PROCESS



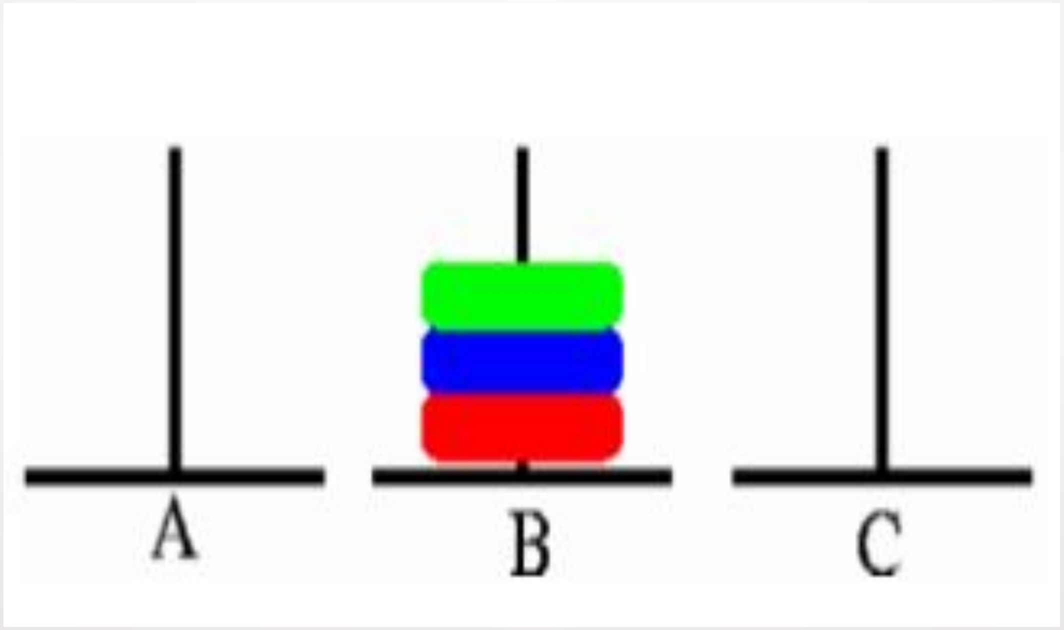
EXAMPLE PROBLEM
PEGS AND DISKS PROBLEM

- Consider the following problem.
  - We have 3 pegs and 3 disks.
  - Operators: one may move the topmost disk on any needle to the topmost position to any other needle.

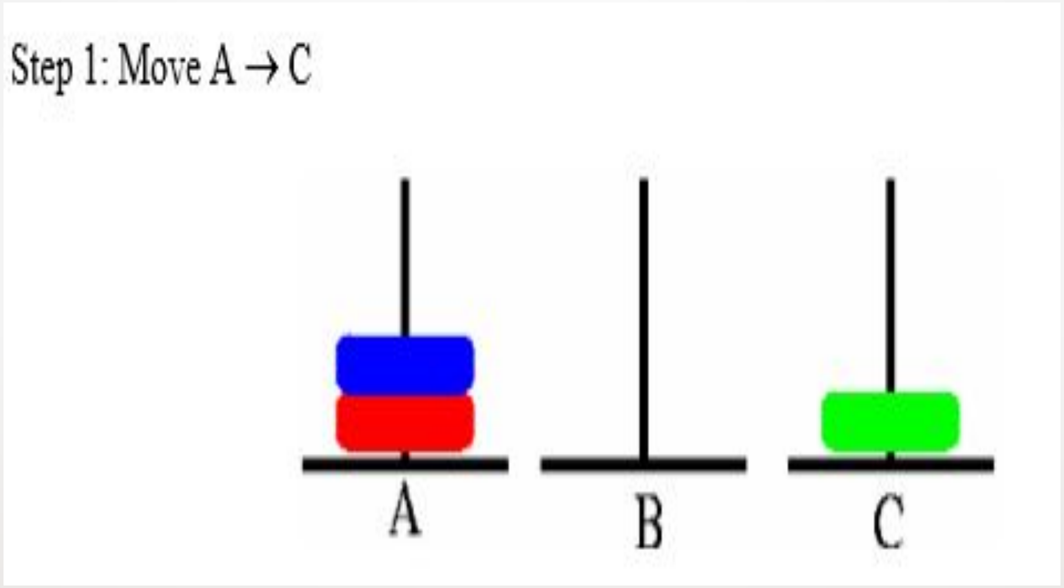
INITIAL STATE



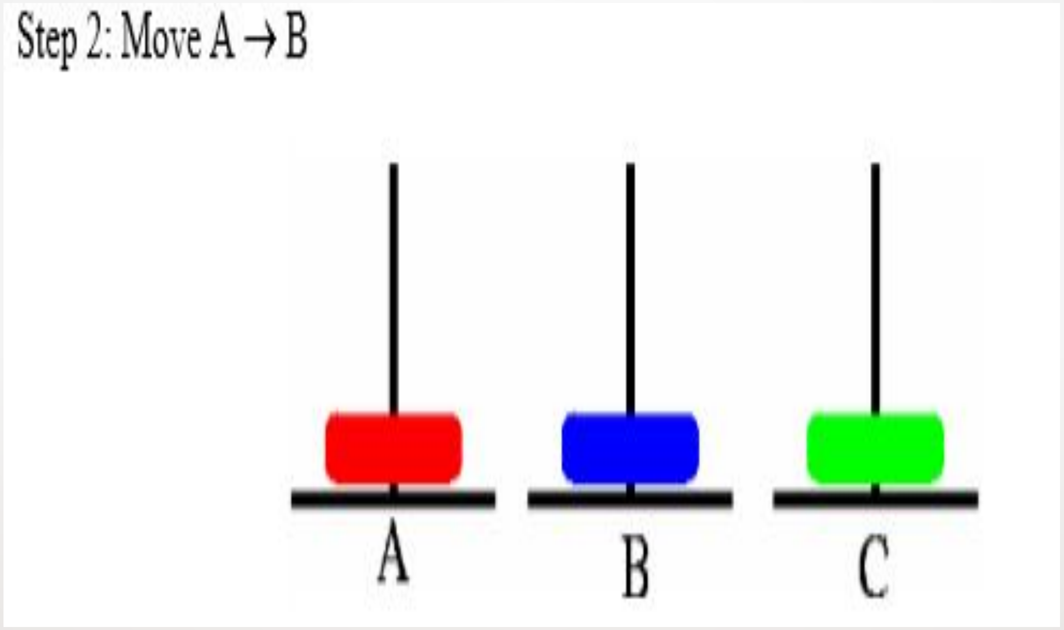
GOAL STATE



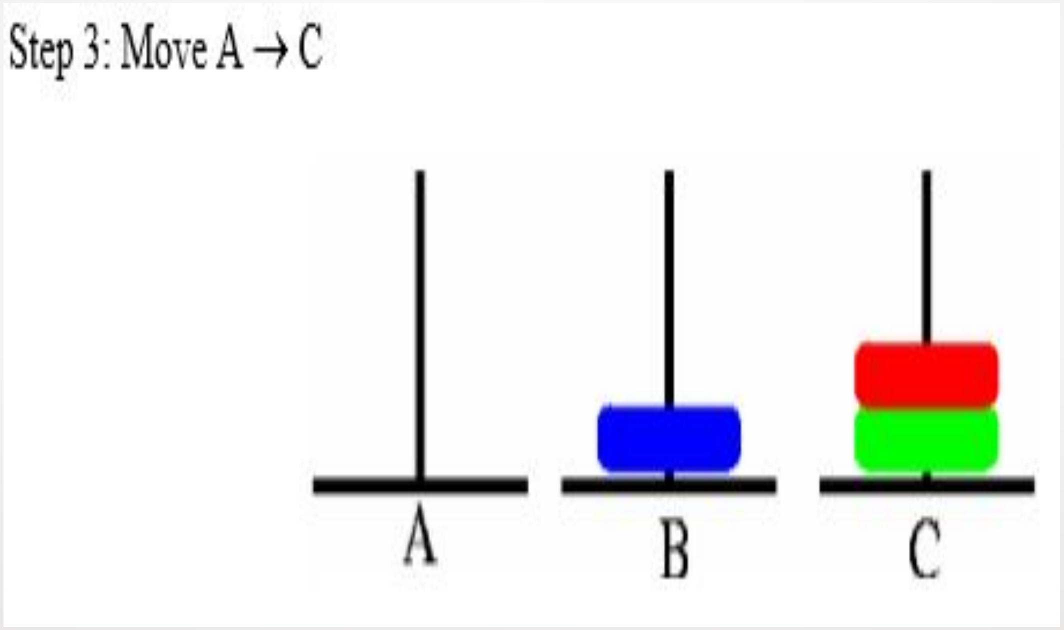
SEQUENCE OF ACTIONS



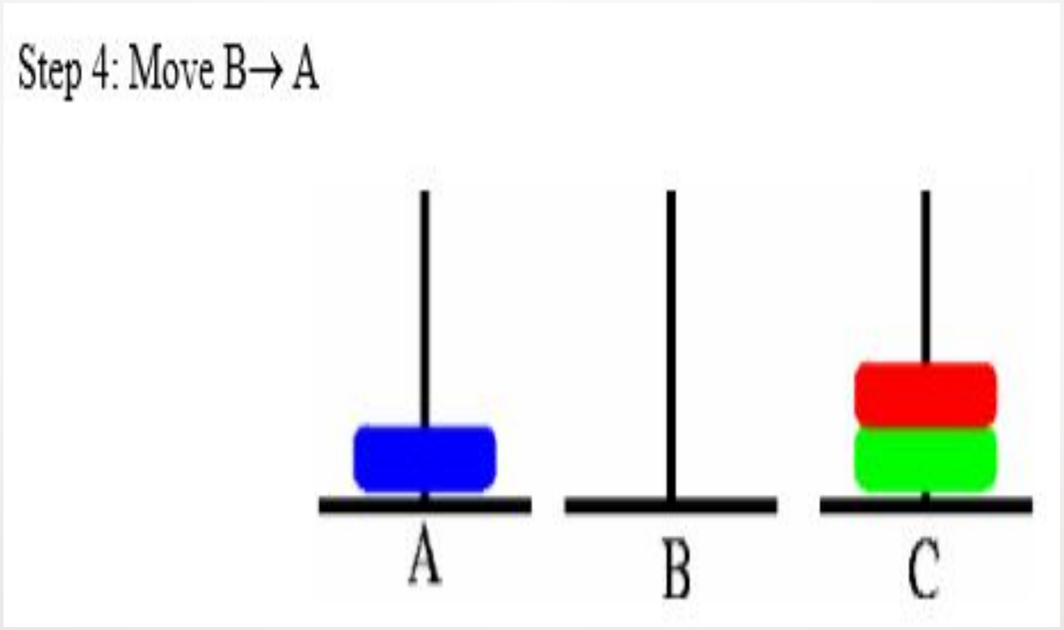
SEQUENCE OF ACTIONS



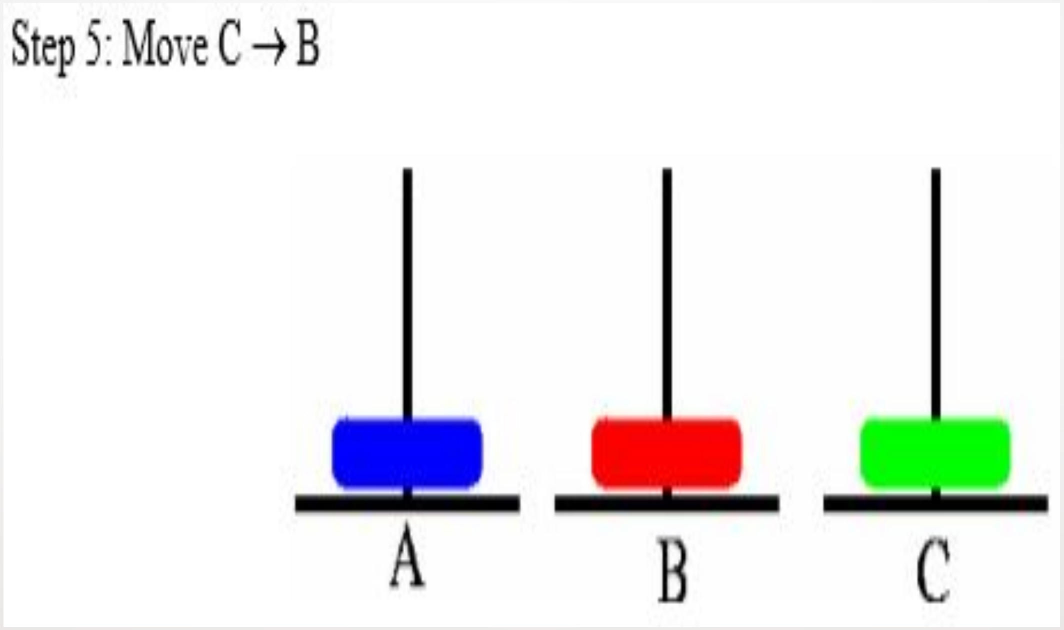
SEQUENCE OF ACTIONS



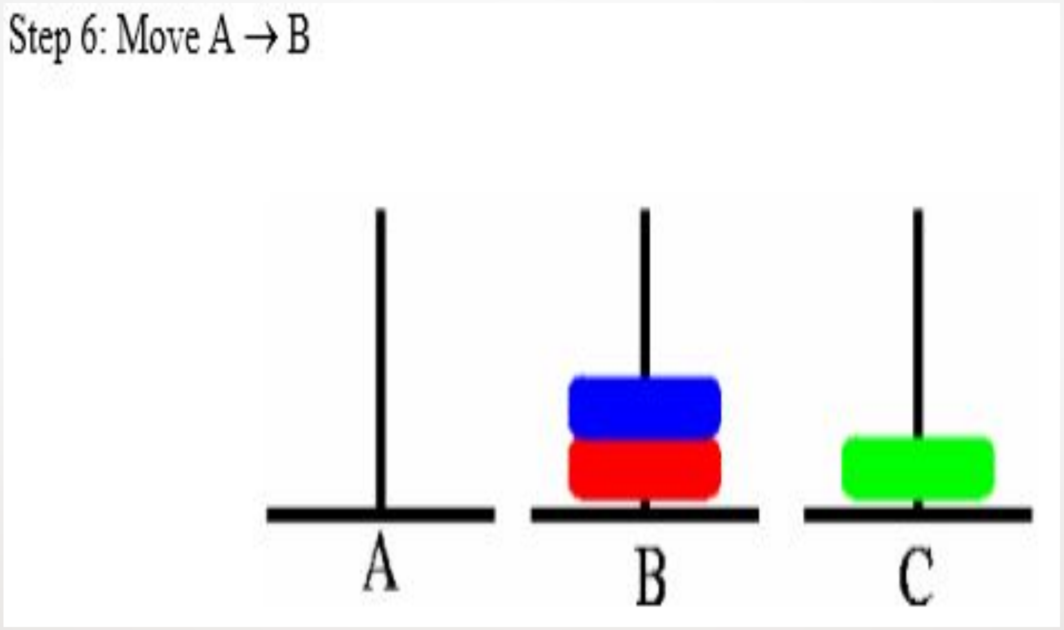
SEQUENCE OF ACTIONS



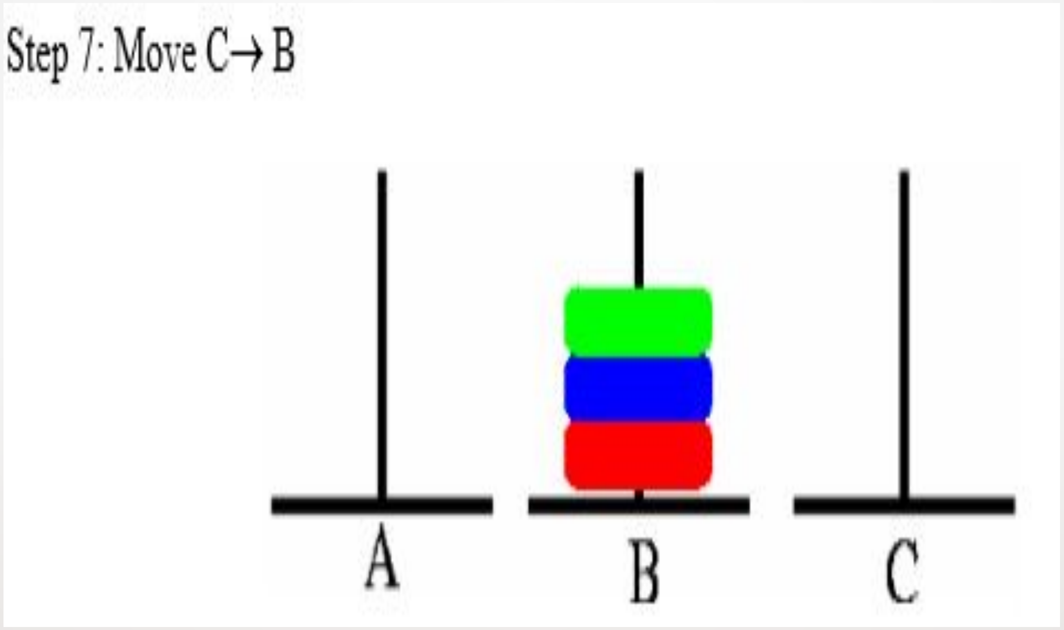
SEQUENCE OF ACTIONS



SEQUENCE OF ACTIONS

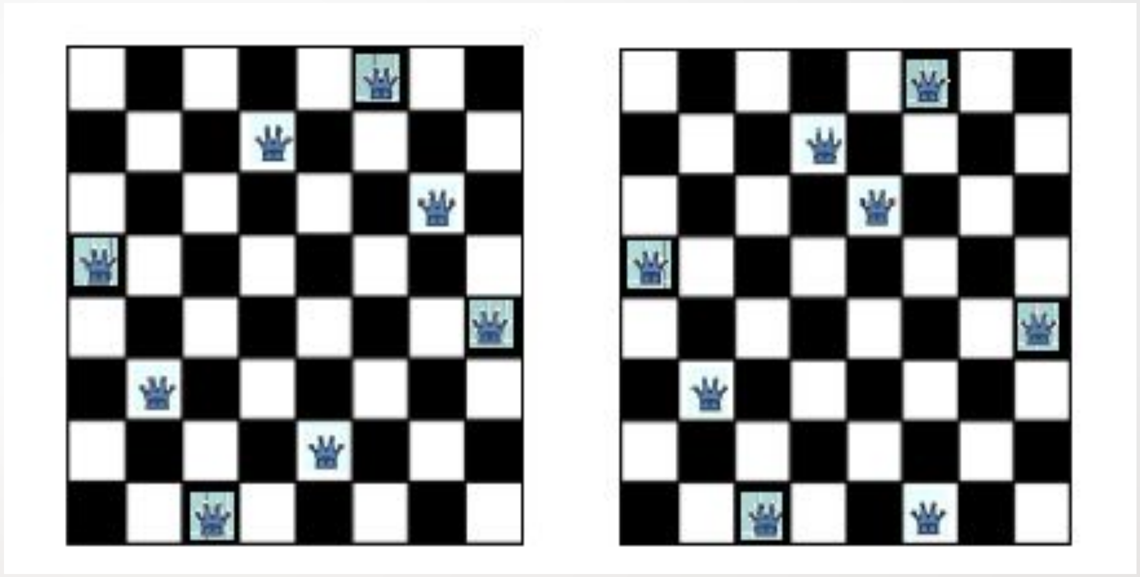


SEQUENCE OF ACTIONS



8 QUEENS PROBLEM

- The problem is to place 8 queens on a chessboard so that no two queens are in the same row, column or diagonal
- The picture on the left shows a solution of the 8-queens problem. The picture on the right is not a correct solution, because some of the queens are attacking each other.



N QUEENS PROBLEM FORMULATION

- Problem formulation
  - Deciding the representation of the states
  - Selecting the initial state representation
  - The description of the operators, and the successor states
- We can formulate the search problem in several different ways for this problem.

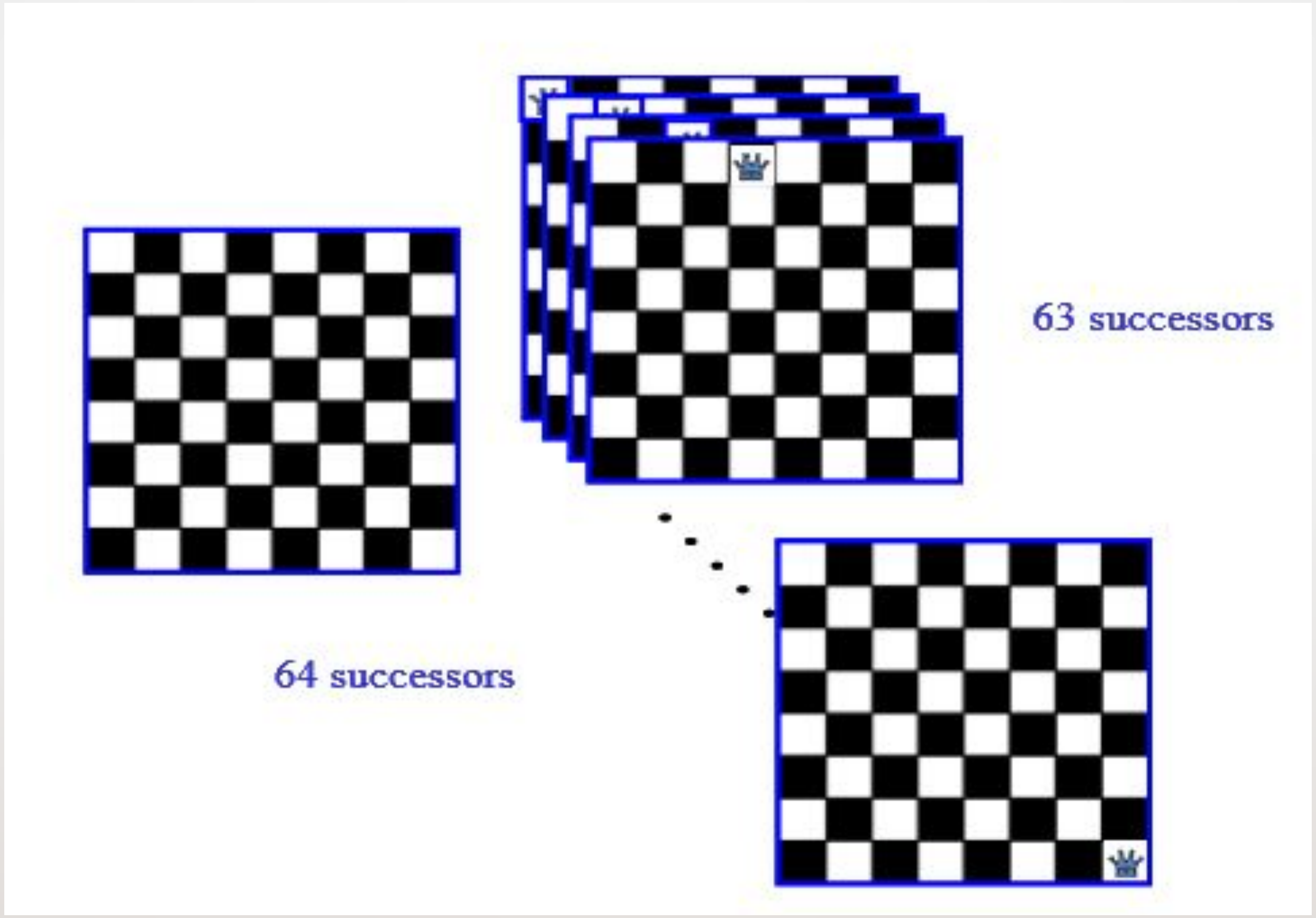
N QUEENS PROBLEM FORMULATION

- N queens problem formulation 1
  - States: Any arrangement of 0 to 8 queens on the board
  - Initial state: 0 queens on the board
  - Successor function: Add a queen in any square
  - Goal test: 8 queens on the board, none are attacked

N QUEENS PROBLEM FORMULATION 1

- The initial state has 64 successors.
- Each of the states at the next level have 63 successors, and so on.
- We can restrict the search tree somewhat by considering only those successors where no queen is attacking each other. To do that we have to check the new queen against all existing queens on the board.

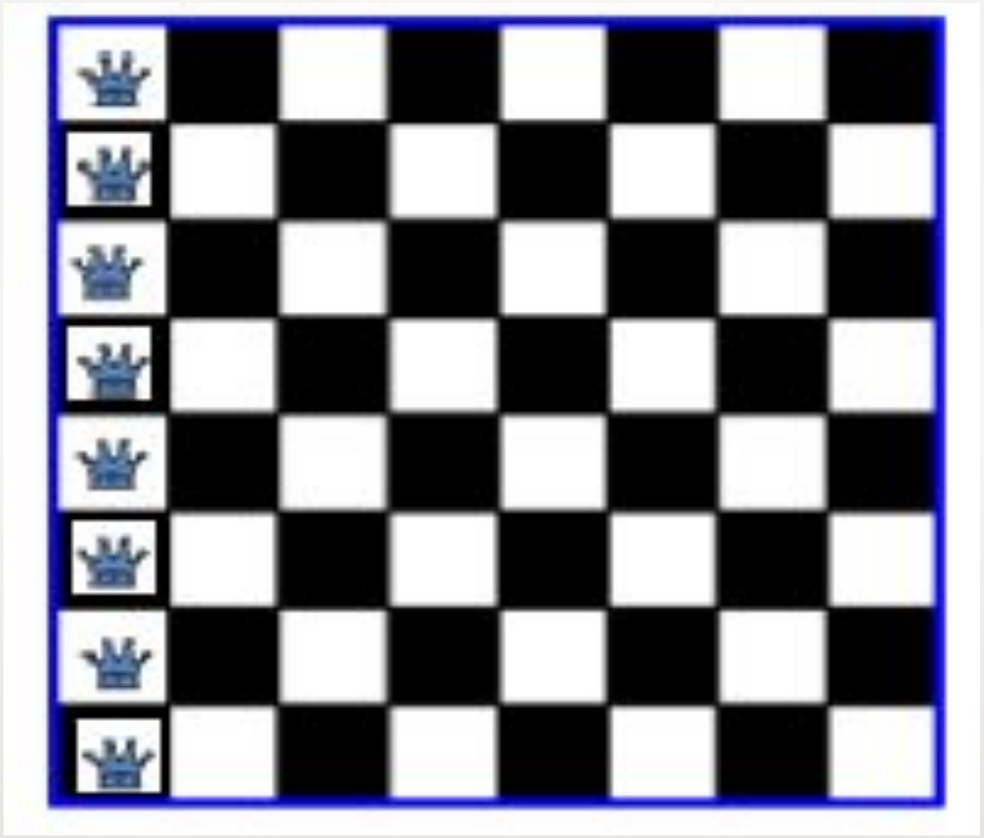
N QUEENS PROBLEM FORMULATION 1



N QUEENS PROBLEM FORMULATION 2

- States:
  - Any arrangement of 8 queens on the board
- Initial state:
  - All queens are at column 1
- Successor function:
  - Change the position of any one queen
- Goal test:
  - 8 queens on the board, none are attacked
- If we consider moving the queen at column 1, it may move to any of the seven remaining columns.

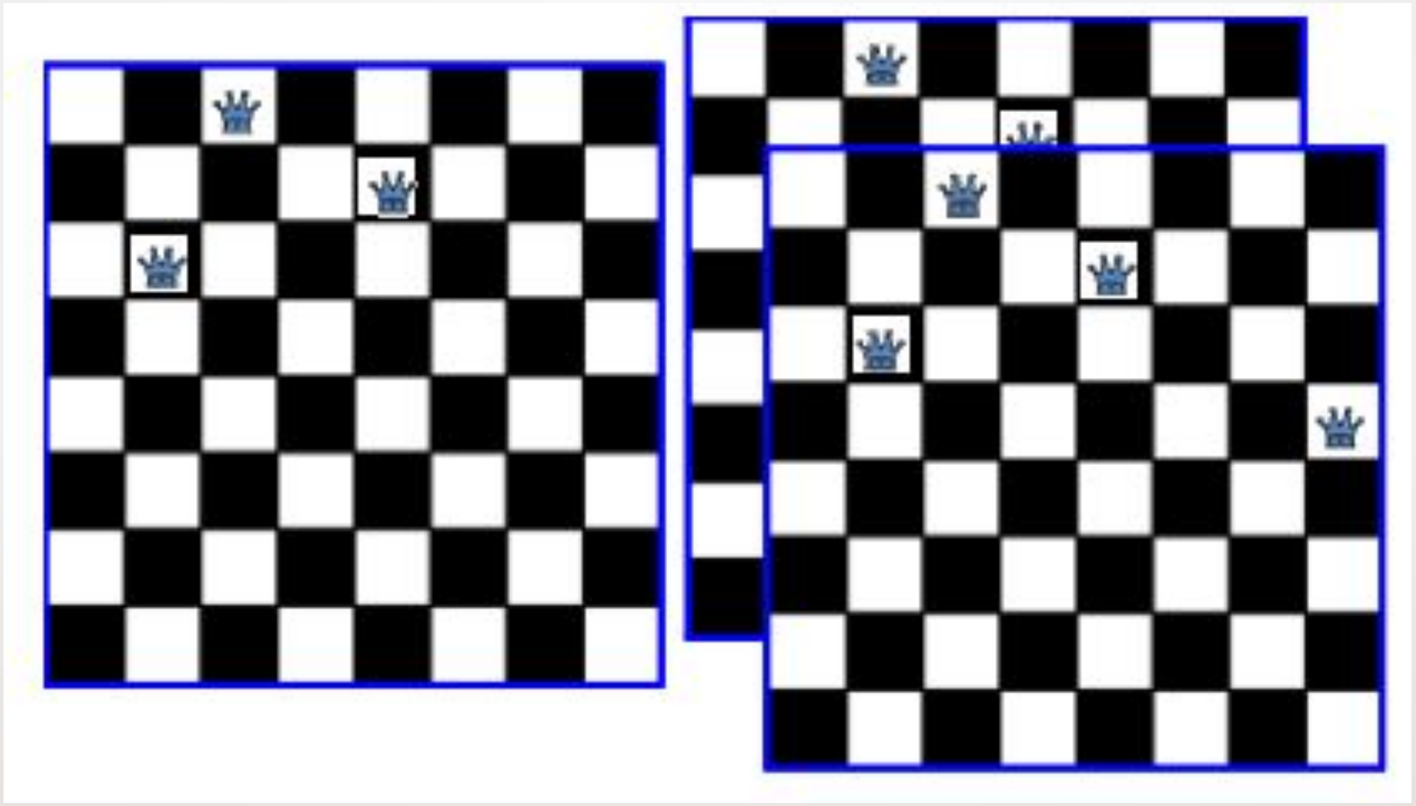
N QUEENS PROBLEM FORMULATION 2



N QUEENS PROBLEM FORMULATION 3

- States:
  - Any arrangement of k queens in the first k rows such that none are attacked
- Initial state:
  - 0 queens on the board
- Successor function:
  - Add a queen to the (k+1)th row so that none are attacked.
- Goal test :
  - 8 queens on the board, none are attacked

N QUEENS PROBLEM FORMULATION 3



Thank You