# Lecture 08
# Problem Formulation

Artificial Intelligence

Dr. Ahmed Mateen

# Today's Agenda

- Problem Solving Agents
- Problem Formulation
- Problem Types
- Examples

# Problem Solving Agents

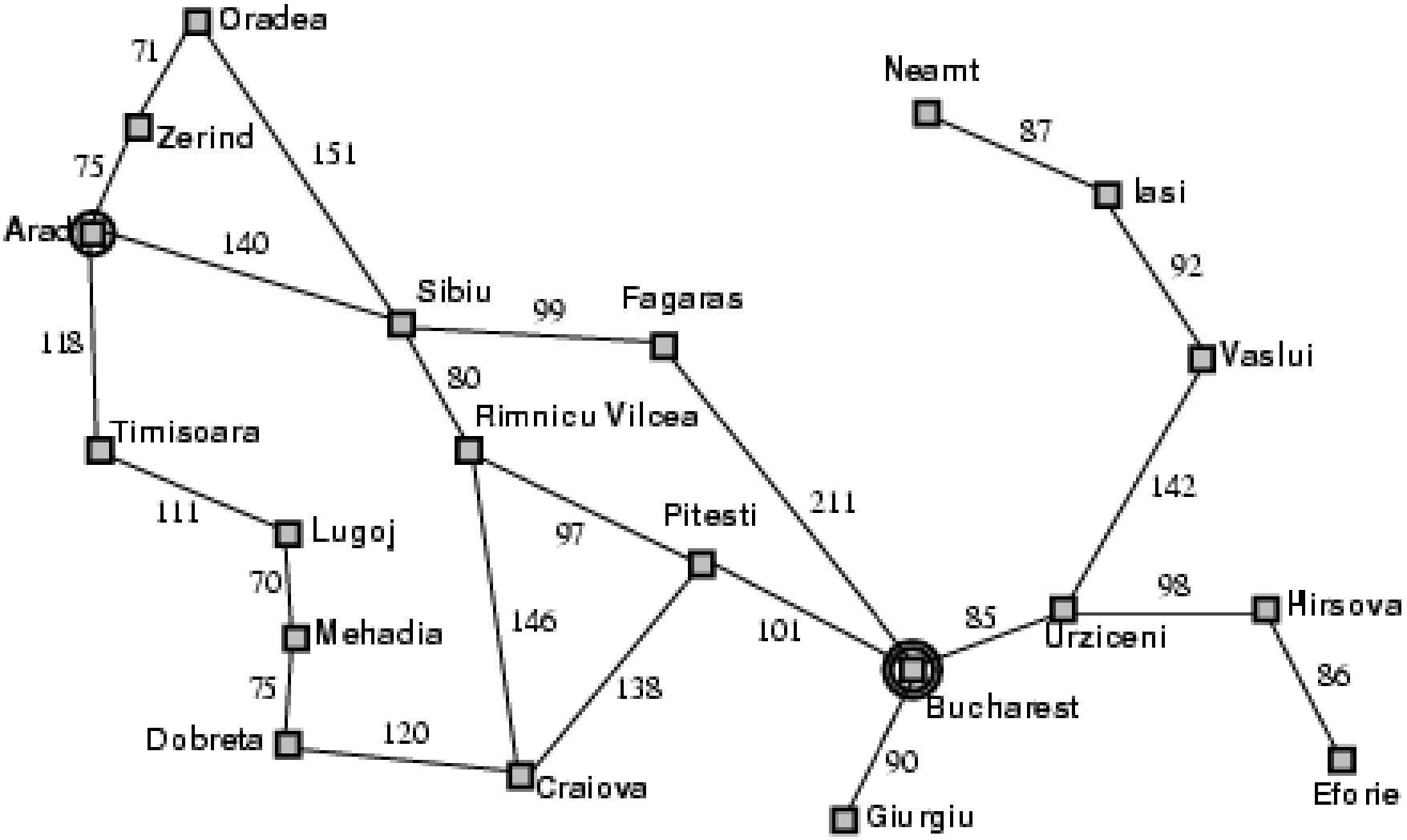
- Problem solving agent
  - A kind of “goal based” agent
  - Finds sequences of actions that lead to desirable states.

- The algorithms are uninformed
  - No extra information about the problem other than the definition
    - No extra information
    - No heuristics (rules)

# Problem Solving Agents

- **Goal formulation**
  - based on the current situation and the agent's performance measure, is the first step in problem solving.
- **Problem formulation**
  - is the process of deciding what actions and states to consider, given a goal

# Example: Romania



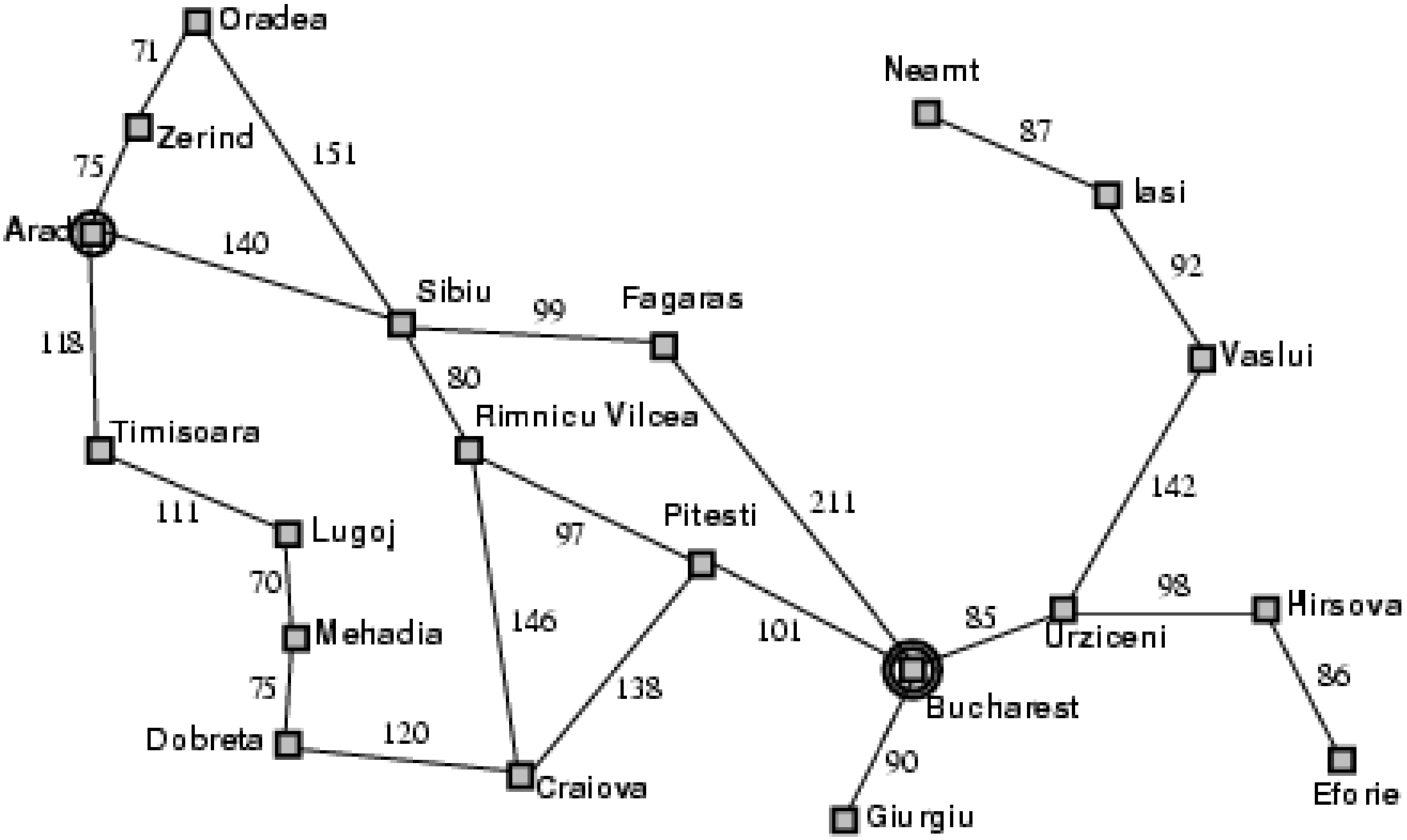
# Important Terminologies

- Transition Model
  - A description of what each action does
- State Space
  - Together, the initial state, actions, and transition model implicitly define the **state space** of the problem
  - the set of all states reachable from the initial state by any sequence of actions
- Successor
  - the term **successor** refers to any state reachable from a given state by a single action.
- Graph
  - The state space forms a directed network or **graph** in which the nodes are states and the links between nodes are **actions**.
- Path
  - A **path** in the state space is a sequence of states connected by a sequence of actions.

# Costs

- Path Cost
- Step Cost
- Optimal Solution

# Example: Romania



# Example: Romania

- On holiday in Romania; currently in Arad.
- Flight leaves tomorrow from Bucharest

- Formulate goal:
  - be in Bucharest

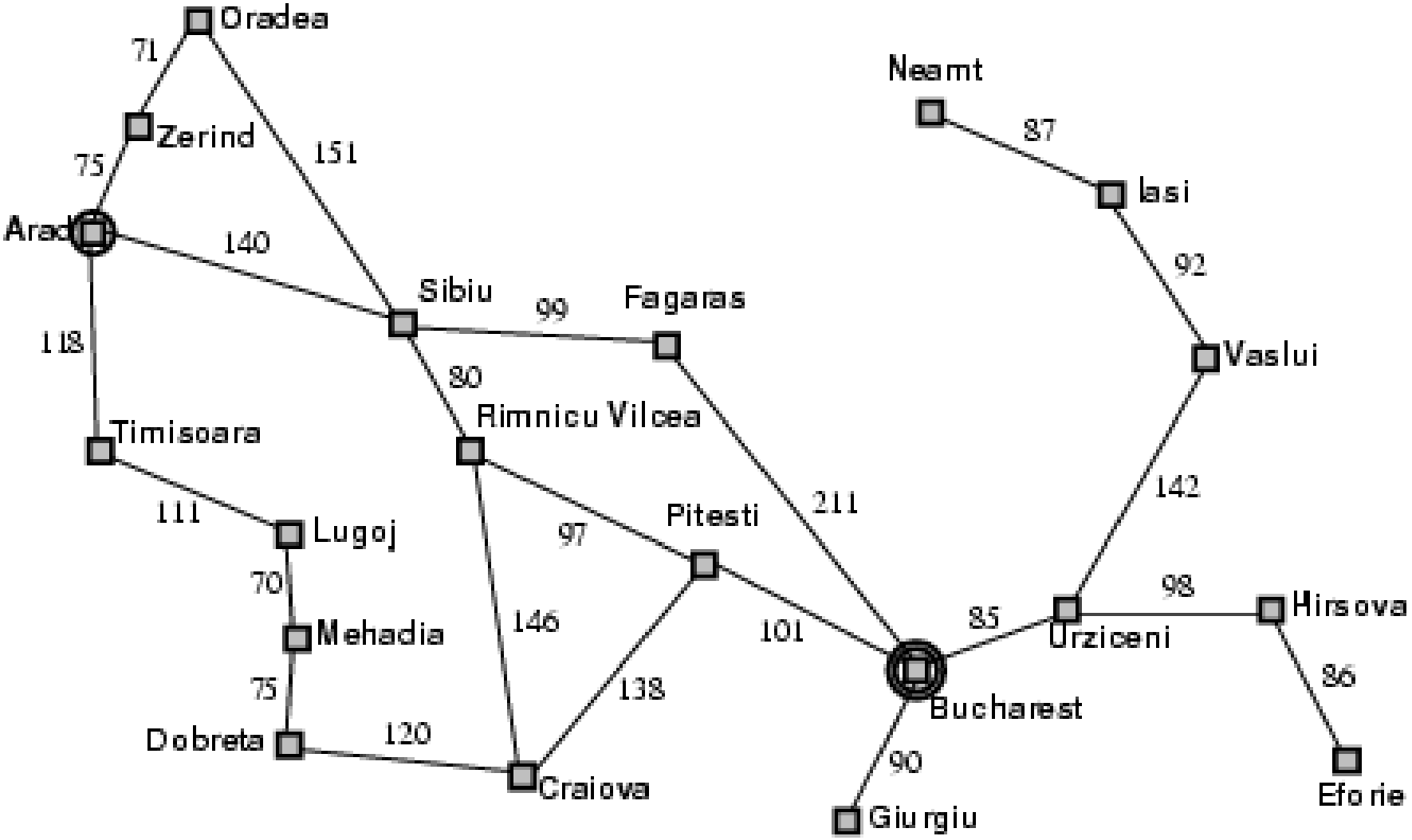
- Formulate problem:
  - states: various cities
  - actions: drive between cities

- Find solution:
  - sequence of cities, e.g., Arad, Sibiu, Fagaras, Bucharest

# Well Defined Problems and Solutions

- A problem
  - Initial state
  - Actions and Successor Function
  - Goal test
  - Path cost

# Example: Romania



# Problem Formulation

- A problem is defined by four items:
- Initial State
  - e.g. “At Arad”
- Successor Function
  - A set of action state pairs
  - S(Arad) = {(Arad->Zerind), …}
- Goal Test
  - e.g. x = “at Bucharest”
- Path Cost
  - sum of the distances traveled
- A solution is a sequence of actions leading from the initial state to a goal state

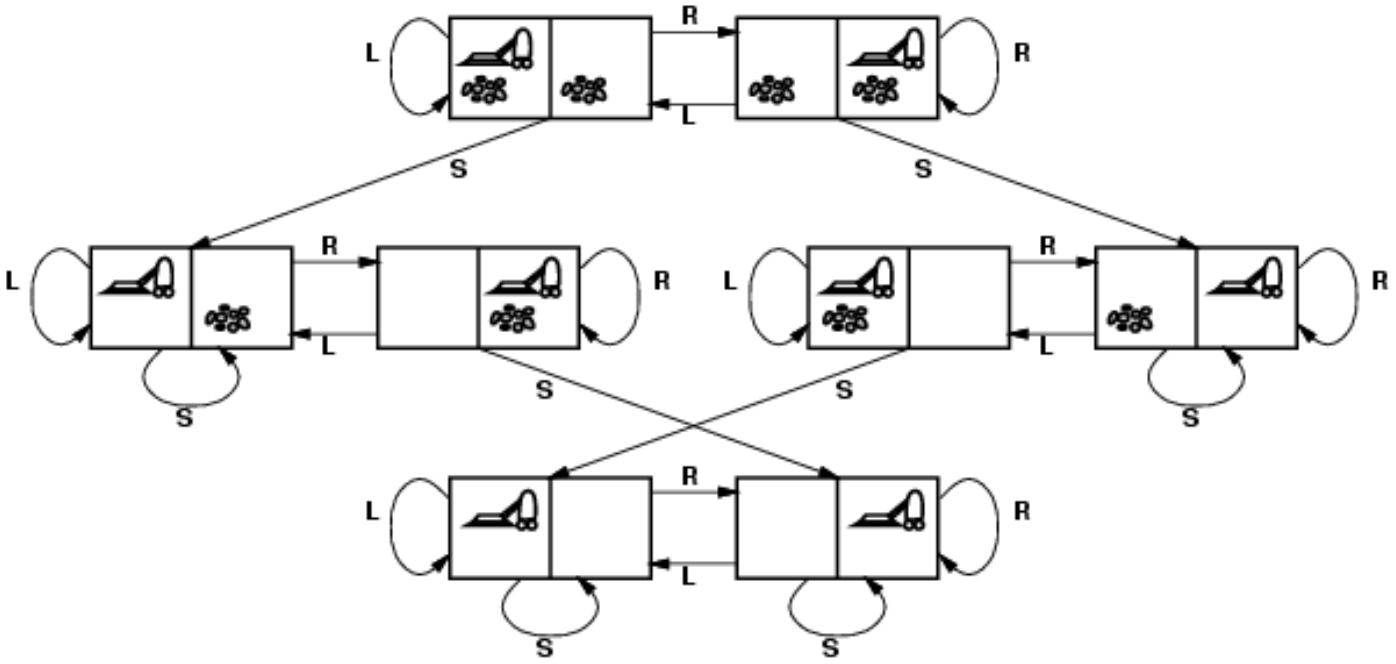
# Problem types

- Deterministic → single-state problem
  - Agent knows exactly which state it will be in; solution is a sequence
- fully observable
  - Agent has access to all information in the environment relevant to its task.
- Discrete → finite locations problem
  - Agent can enumerate choices
- Static
  - The plan remains the same

# Problem types Contd…

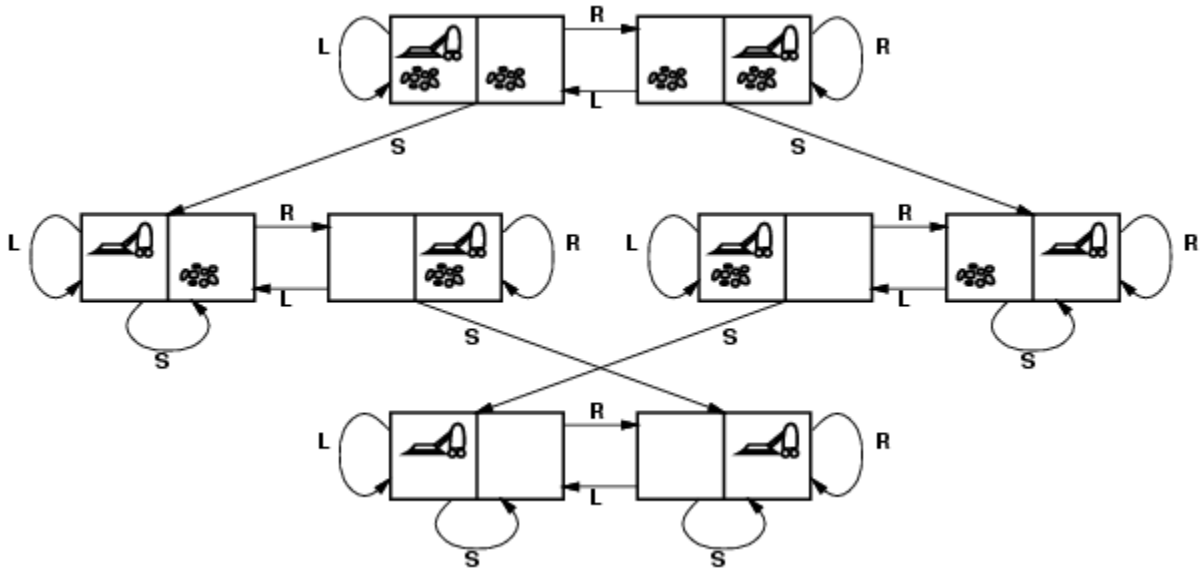
- Non-observable → sensorless problem (conformant problem)
  - Agent may have no idea where it is; solution is a sequence
- Nondeterministic and/or partially observable → contingency problem
  - percepts provide new information about current state
- Unknown state space → exploration problem

# Vacuum world state space graph



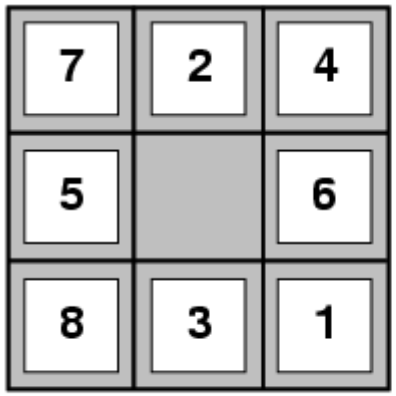

- states?
- actions?
- goal test?
- path cost?

# Vacuum world state space graph

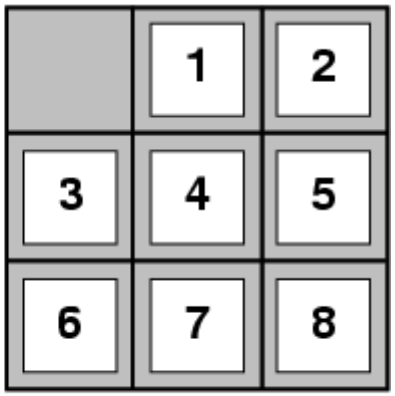



- states? Dirt and robot location
- actions? *Left*, *Right*, *Suck*
- goal test? no dirt at all locations
- path cost? 1 per action

# Example: The 8-puzzle



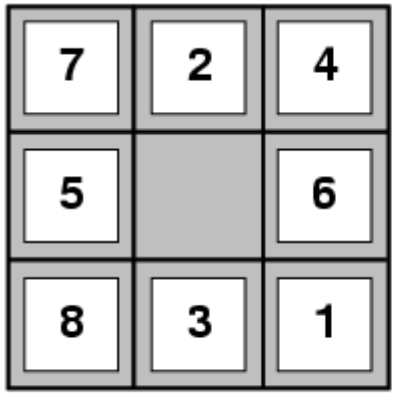
Start State



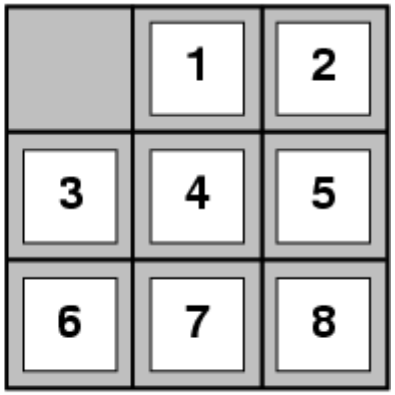
Goal State

- states?
- actions?
- goal test?
- path cost?

# Example: The 8-puzzle




Start State

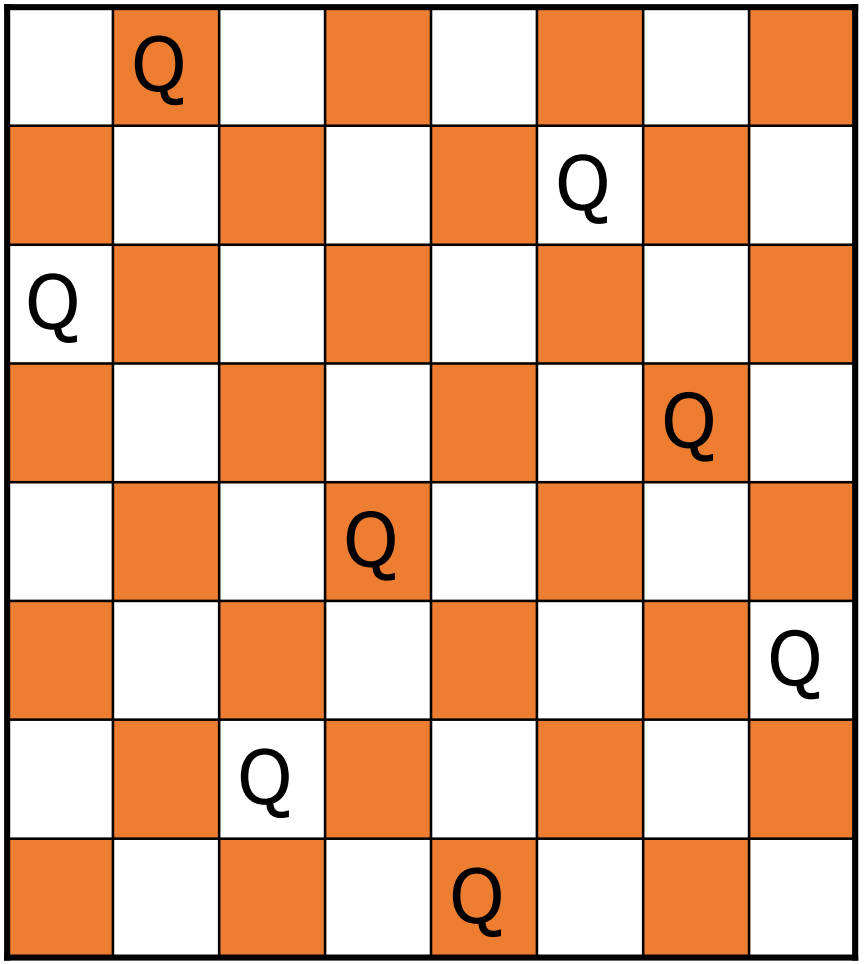



Goal State

- states? locations of tiles
- actions? move blank left, right, up, down
- goal test? = goal state (given)
- path cost? 1 per move

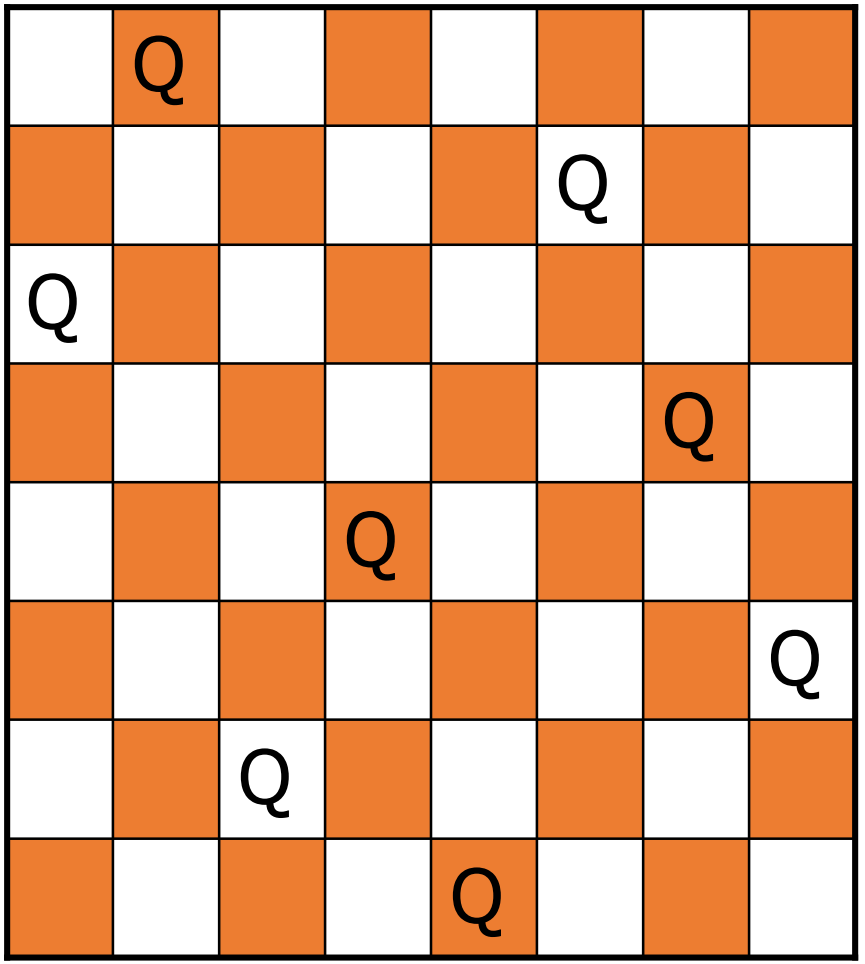
# Example: Eight Queens

- Place eight queens on a chess board such that no queen can attack another queen
- No path cost because only the final state counts!
- Incremental formulations
- Complete state formulations



# Example: Eight Queens

- States:
  - Any arrangement of 0 to 8 queens on the board
- Initial state:
  - No queens on the board
- Successor function:
  - Add a queen to an empty square
- Goal Test:
  - 8 queens on the board and none are attacked
- 64*63*…*57 = $1.8*10^{14}$ possible sequences
  - Ouch!



# Example: Eight Queens

- States:
  - Arrangements of n queens, one per column in the leftmost n columns, with no queen attacking another are states
- Successor function:
  - Add a queen to any square in the leftmost empty column such that it is not attacked by any other queen.