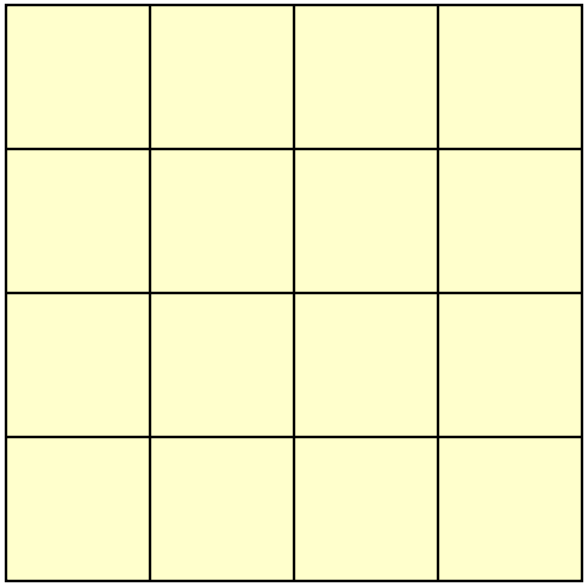
Problem-solving using search

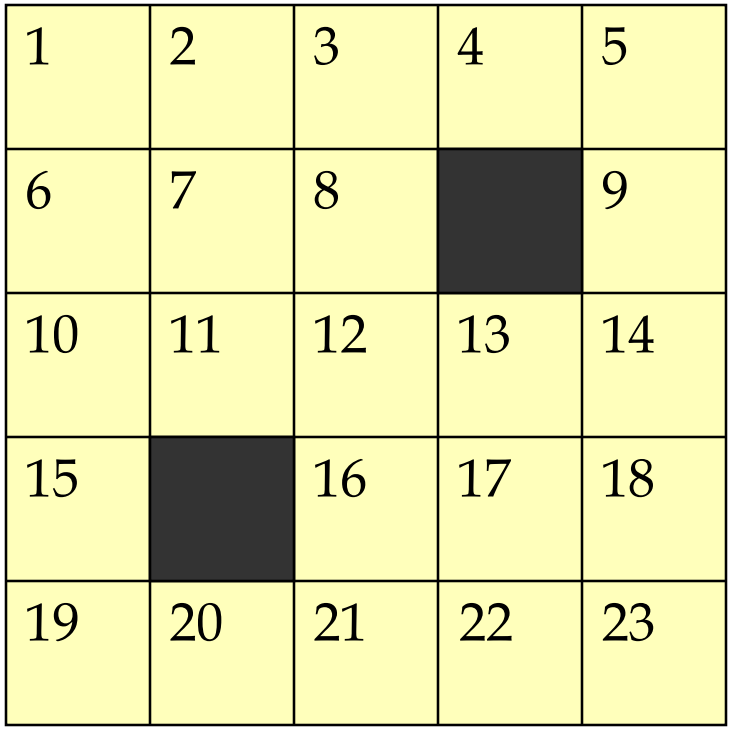
Problem-solving agents: Goal-based agents that decide what to do by finding sequences of actions that lead to desirable states.

Example: *n*-queens

Place *n*-queens on an $n \times n$ board so that no pair of queens attacks each other.



Example: crossword puzzles

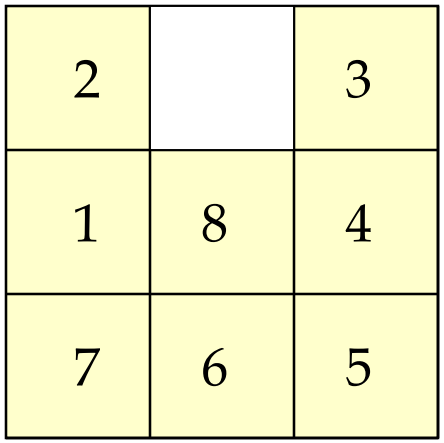


a
aardvark
aback
abacus
abaft
abalone
abandon
...

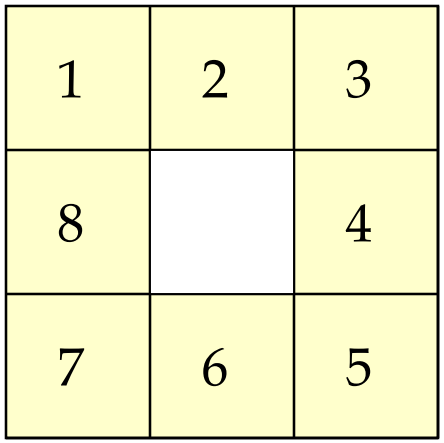
…
monarch
monarchy
monarda
...
zymurgy
zyrian
zythum

Example: sliding puzzles

Initial configuration



Goal configuration



Example: river crossing puzzle

A father, his two sons, and a boat are on one side of a river. The capacity of boat is 100 kg. The father weighs 100 kg and each son weighs 50 kg. How can they get across the river?

Example: propositional satisfiability

Given a formula in propositional logic, determine if the Boolean variables can be assigned in such a way as to make the formula true.

$$
\begin{aligned}
&(\neg A \vee B) \wedge \\
&(\neg B \vee \neg C \vee D) \wedge \\
&(\neg D \vee G \vee \neg E) \wedge \\
&(\neg D \vee G \vee \neg F) \wedge \\
&A \wedge \\
&C \wedge \\
&\neg E
\end{aligned}
$$

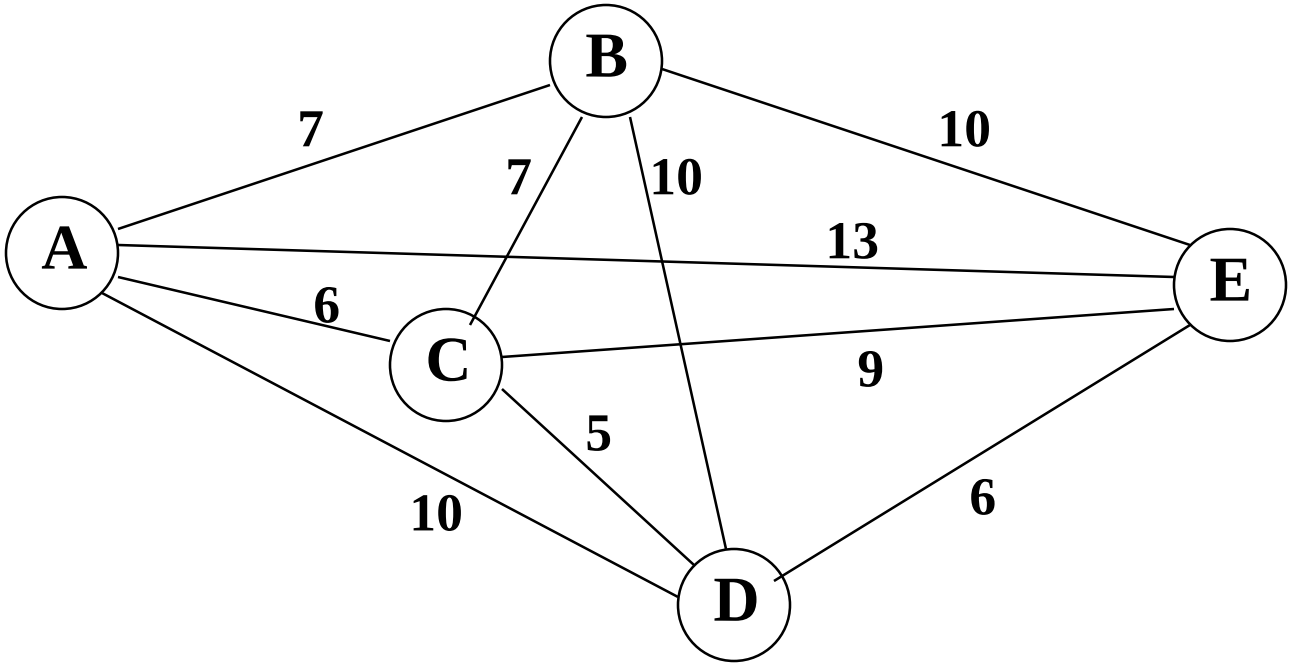
Example: partition problem

Given a set of objects with weights, partition the objects into two sets *U* and *V* such that the total weights of *U* and *V* are as close as possible.

Object	a	b	c	d	e	f	g	h
Weight	5	7	10	10	11	15	16	16

Example: traveling saleswoman problem

Starting at city A, find a route of minimal distance that visits each of the cities only once and returns to A.



Example: set covering problem

Find a minimum size committee of people that together have the skills necessary to accomplish a task.

SkillsNeeded = {a, b, c, d, e, f, g, h, i, j, k, l}

People = {p_1, p_2, p_3, p_4, p_5, p_6}, where

p_1 has skills {a, b, e, f, i, j}

p_2 has skills {f, g, j, k}

p_3 has skills {a, b, c, d}

p_4 has skills {c, e, f, g, h}

p_5 has skills {i, j, k, l}

p_6 has skills {d, h}

Contrasts

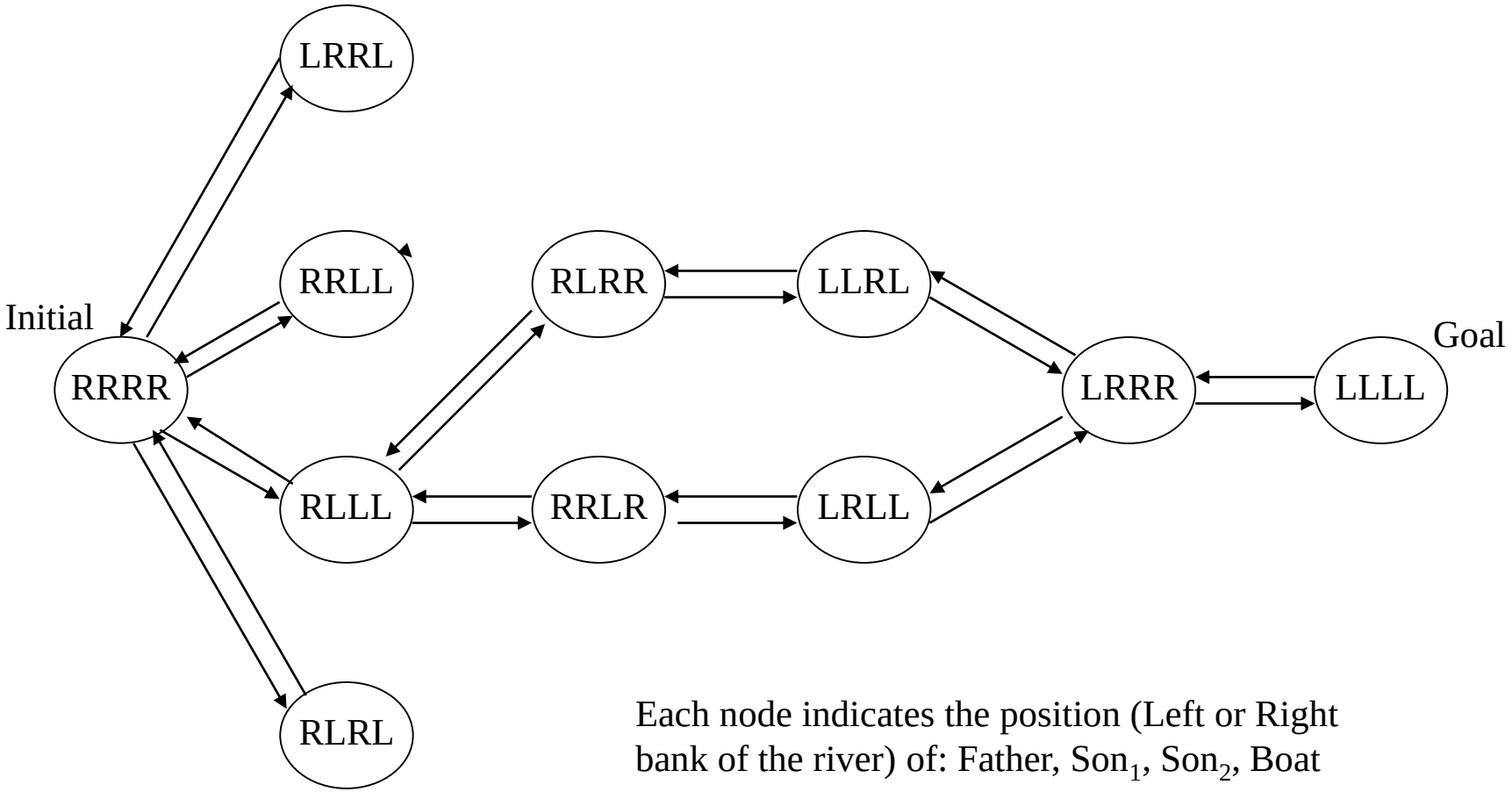
- Find a goal state, given constraints on the goal
  - any goal state
    - e.g., *n*-queens, crossword puzzles
  - *optimal* goal state
    - e.g., traveling saleswoman problem, set covering problem
- Find a sequence of actions that lead to goal state
  - any sequence
    - e.g., sliding puzzle, river crossing puzzle
  - *optimal* sequence
    - e.g., sliding puzzle, …

Methodology

- Formulate problem solving as search on a graph
- Given a problem:

nodes	1. Specify representation of states, specify initial and goal states
arcs	2. Specify actions (successor function, neighborhood function) that take us from one state to another. Assign a cost to each action
search	3. Find a path from an initial state to a goal state

Search graph for River Crossing Puzzle



Each node indicates the position (Left or Right bank of the river) of: Father, Son_1, Son_2, Boat

General search algorithm

L ← [start nodes]
while L ≠ empty do
    select and remove a node from L, call it p
    if p is a goal node, return(success)
    generate all successor states of p, and add them to L
return(fail)

FIFO queue gives Breadth-First Search (BFS)
LIFO queue gives Depth-First Search (DFS)
Priority queue gives informed search (greedy, A*)

What to do about repeated states?

0. Nothing
1. Don’t return to a state that you just came from
2. Do not create paths with cycles in them (look at ancestors of a node)
3. Do not generate any state that was ever generated before (keep a closed list using a hash table)