Search
Introduction and Problem Formulation

Alice Gao

Lecture 2

Based on work by K. Leyton-Brown, K. Larson, and P. van Beek

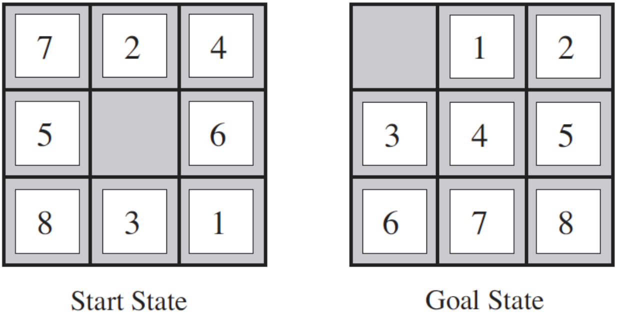
Outline

Learning goals

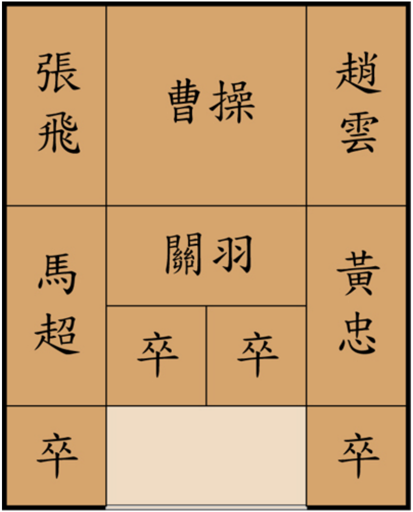
By the end of the lecture, you should be able to

- Describe the components of a search problem.
- Formulate a real world problem as a search problem, using the incremental or the complete-state formulation.
- Given a successor function, give all the successor states of a given state.
- Given a search problem, estimate the complexity of the search space in terms of number of nodes and paths in the search tree.

Example: Sliding puzzles



Example: Hua Rong Pass Puzzle



Example: Rubik's cube

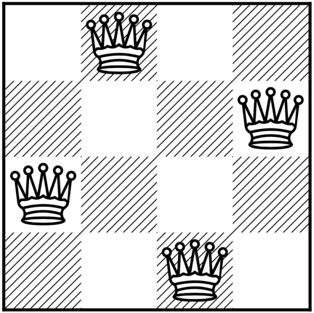
Example: River Crossing Puzzle

A parent and two children are trying to cross a river using a boat.

- The capacity of the boat is 100kg.
- The parent weighs 100kg.
- Each child weighs 50kg.

How can they get across the river?

Example: *N*-Queens Problem



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

Example: Propositional Satisfiability

Given a formula in propositional logic, determine if there is a way to assign truth values to the Boolean variables to make the formula true.

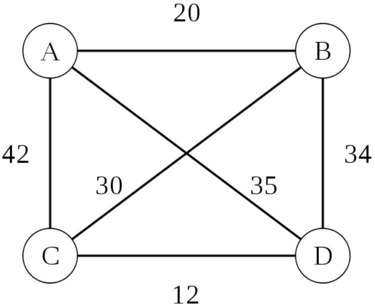
$$((((a \wedge b) \vee c) \wedge d) \vee (\neg e))$$

Applications:

- FCC spectrum auction
- Circuit design
- Planning in AI

Example: Traveling Salesperson Problem

What is the shortest path that starts at city A, visits each city only once, and returns to A?



Applications of TSP: https://bit.ly/2i9JdIV

CQ: Why search?

CQ: Why do we use search to solve these problems? Are there efficient algorithms to solve them? (All answers are correct.)

(A) We use search because there aren't efficient algorithms to solve some of these problems.

(B) We use search for another reason.

(C) Search and efficient algorithms are both valid approaches to solve these problems.

(D) We use search because it works better than efficient algorithms.

Search

We would like to find a solution when we are

- Not given an algorithm to solve a problem
- Given a specification of what a solution looks like
- (Given costs associated with certain actions)

Idea: search for a solution (with the minimum cost)

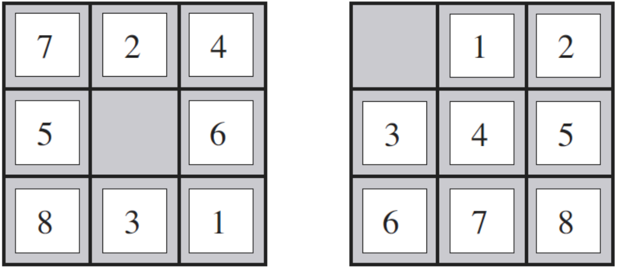
A Search Problem

Definition (Search Problem)

A search problem is defined by

- A set of states
- A start state
- A goal state or goal test
  - a boolean function which tells us whether a given state is a goal state
- A successor function
  - a mapping/action which takes us from one state to other states
- A cost associated with each action

Example: 8-Puzzle



Start State Goal State

CQ: 8-Puzzle

CQ: Which one is NOT a successor of 724560831?

(A) 724563801

(B) 724561830

(C) 720564831

(D) 724506831

CQ: Search Graph for 8-Puzzle

CQ: Your friend is implementing a search algorithm to solve the 8-puzzle. They decide to generate and store the search graph in memory, before running a search algorithm on it.

Do you think this is a reasonable approach?

(A) Yes, this approach sounds reasonable.

(B) No, this approach has serious problems.

Example: River Crossing Puzzle

A parent and two children are trying to cross a river using a boat.

- The capacity of the boat is 100kg.
- The parent weighs 100kg.
- Each child weighs 50kg.

How can they get across the river?

CQ: River Crossing Puzzle

CQ: How many goal states are there for the river crossing puzzle?

(A) 0

(B) 1

(C) 2

(D) 3

(E) More than 3

CQ: River Crossing Puzzle

CQ: Which state has more than one successors?

(A) 1001

(B) 0111

(C) 0101

(D) 0011

An exercise for you

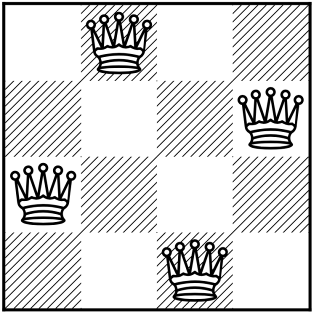
Draw the search graph of the river crossing puzzle.
It should contain $2^4 = 16$ states.

Some questions to think about:

- Which nodes in the search graph are unreachable from any state?
- Which nodes are unreachable if we start at 0000, go through any path and stop when we reach any goal state?

4-Queens Problem

Place 4 queens on an 4×4 board so that no pair of queens attacks each other.



CQ: Incremental Formulation A

CQ: For incremental formulation A,
how many paths do we need to consider in the worst case?

(A) Less than 10^2
(B) Between 10^2 and 10^3
(C) Between 10^3 and 10^4
(D) More than 10^4

CQ: Incremental Formulation B

CQ: For incremental formulation B,
how many paths do we need to consider in the worst case?

(A) Less than 10^2

(B) Between 10^2 and 10^3

(C) Between 10^3 and 10^4

(D) More than 10^4

Revisiting the learning goals

By the end of the lecture, you should be able to

- Describe the components of a search problem.
- Formulate a real world problem as a search problem, using the incremental or the complete-state formulation.
- Given a successor function, give all the successor states of a given state.
- Given a search problem, estimate the complexity of the search space in terms of number of nodes and paths in the search tree.