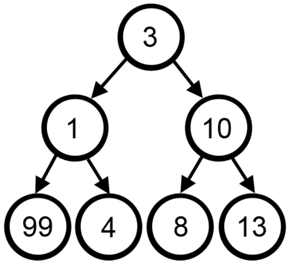
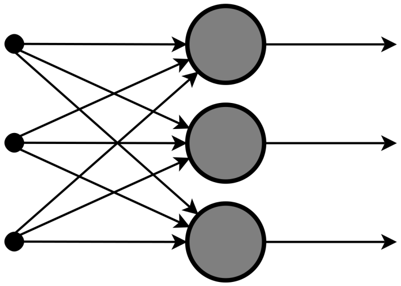
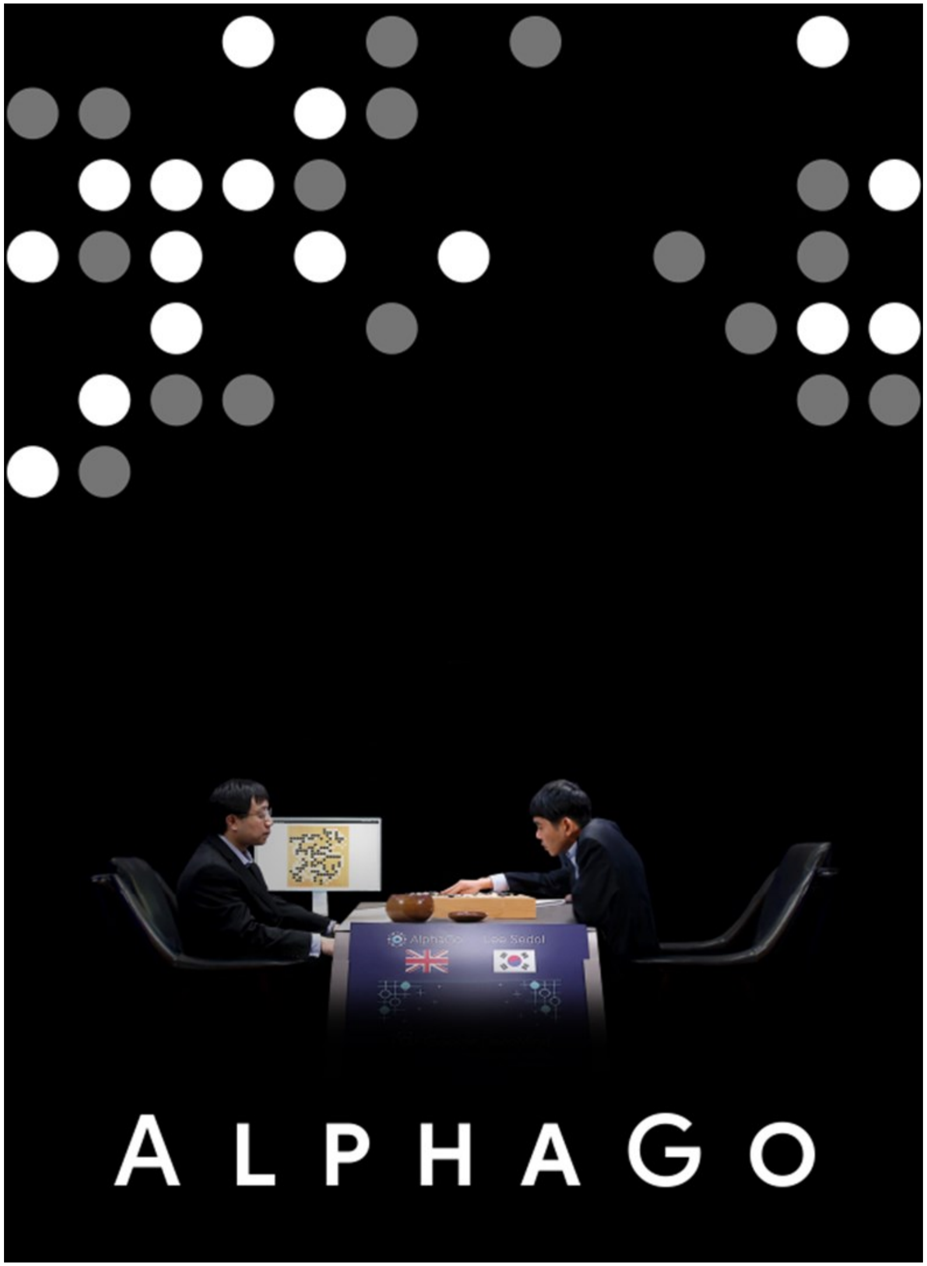
CSE 40171:
Artificial Intelligence





Uninformed Search: Search Spaces

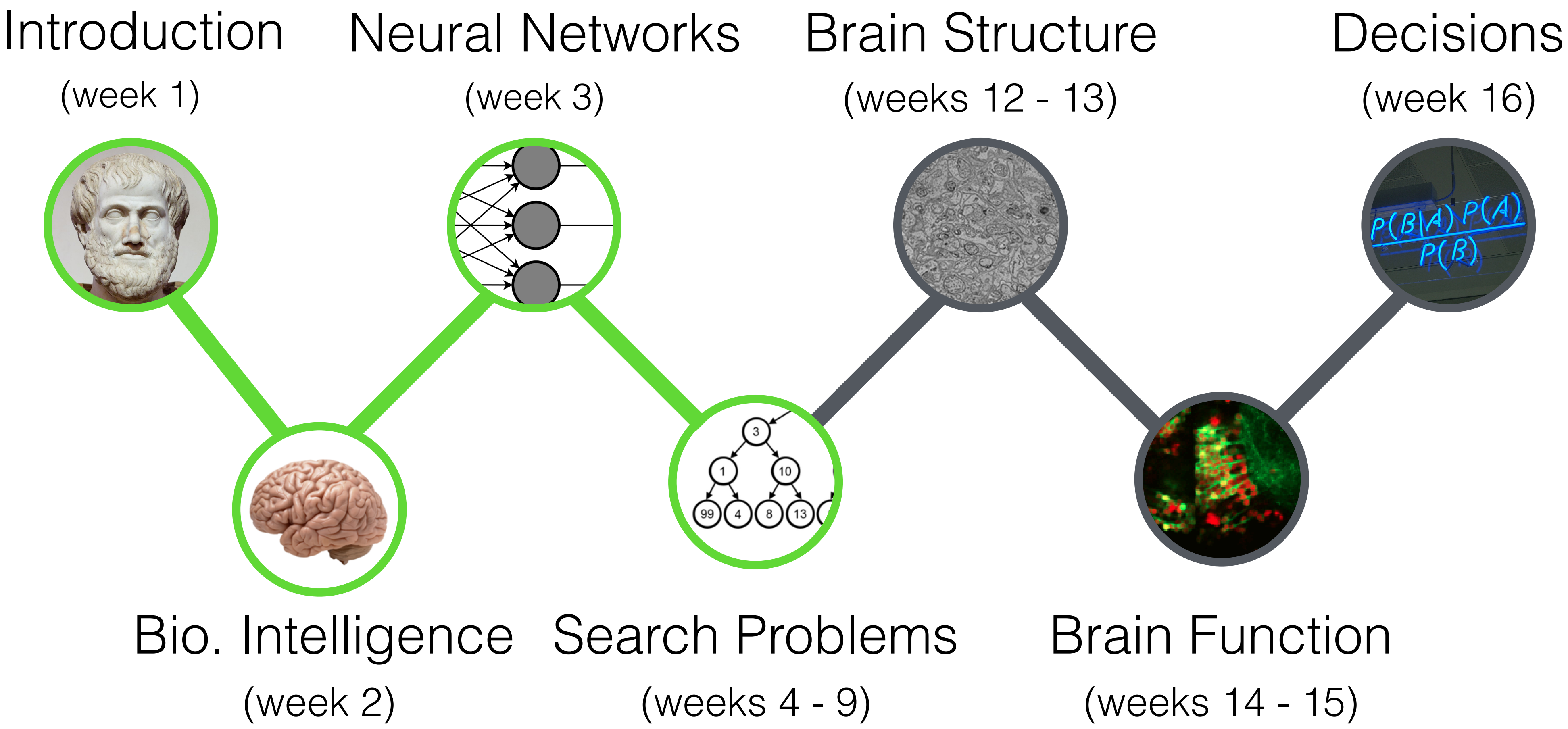
Homework #1 is due **tonight** at 11:59PM



Film Screening:
Wednesday and Friday

Film Response Activity
Due: 9/23

Course Roadmap



Introduction
(week 1)

Bio. Intelligence
(week 2)

Neural Networks
(week 3)

Search Problems
(weeks 4 - 9)

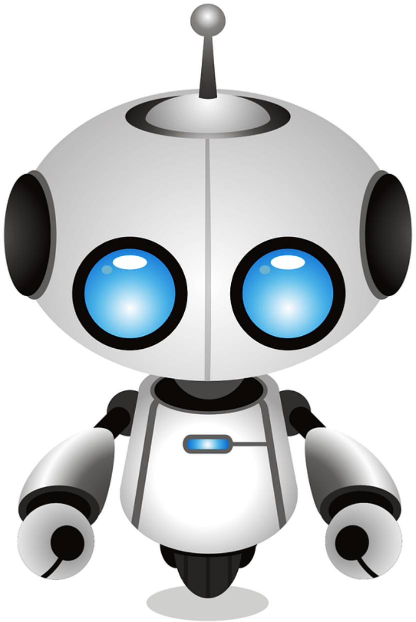
Brain Structure
(weeks 12 - 13)

Brain Function
(weeks 14 - 15)

Decisions
(week 16)

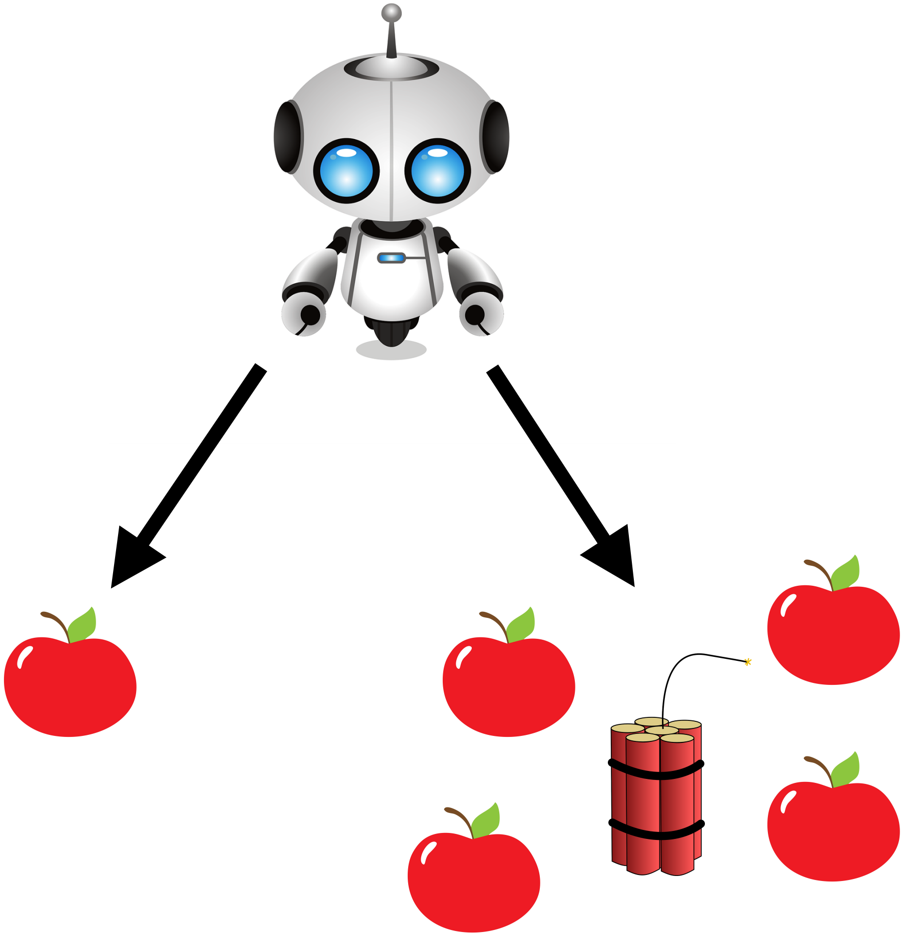
Agents

Definition: anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**

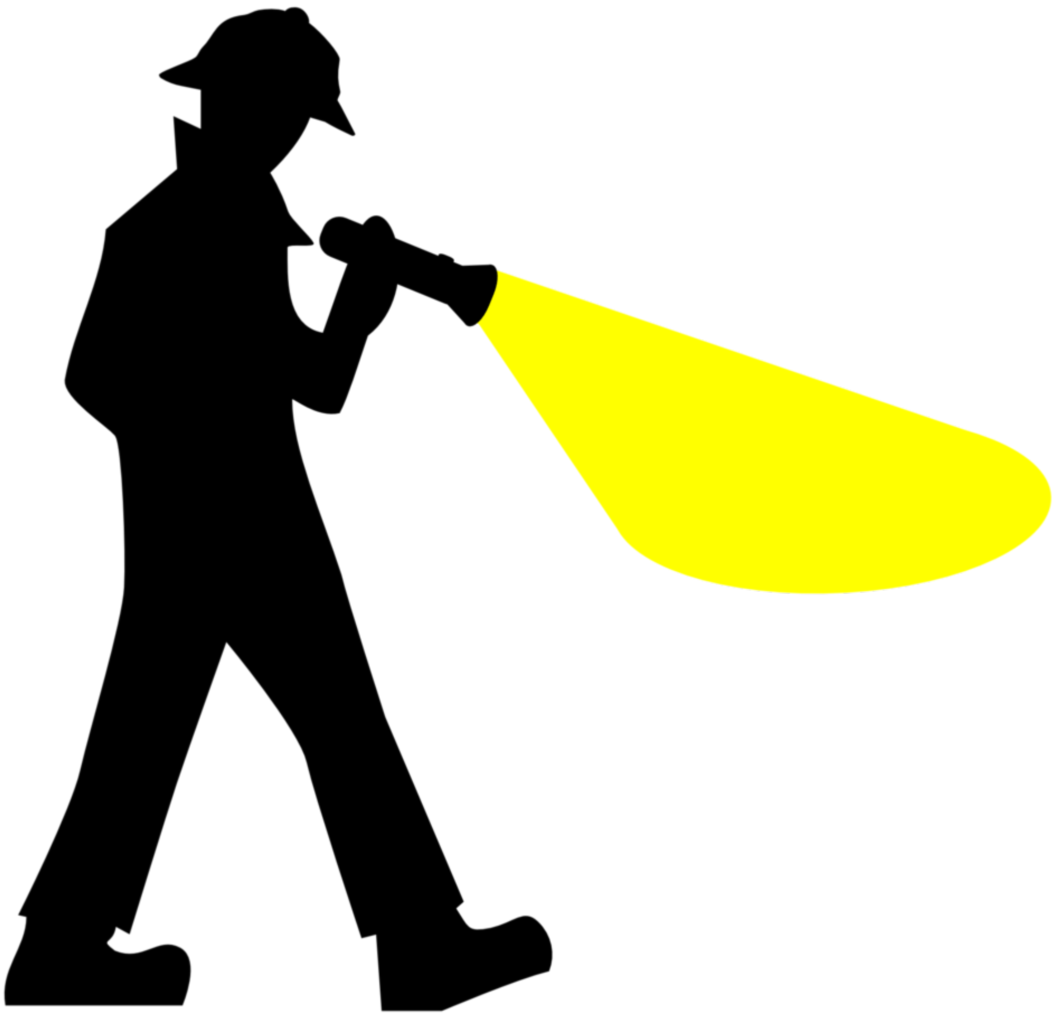


Goal-Based Agents

Consider future actions and the desirability of their outcomes



Search Problems



Touring Romania

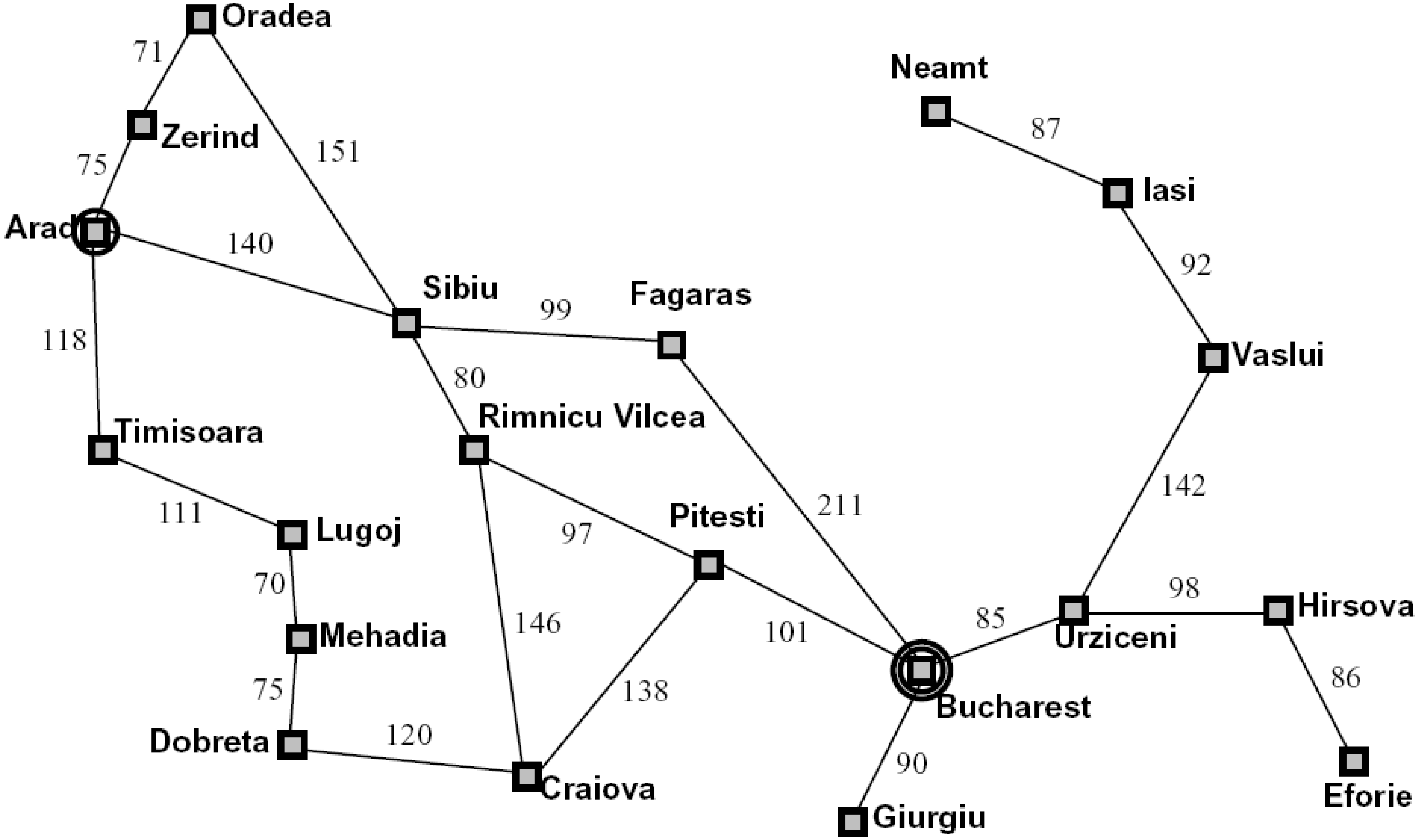


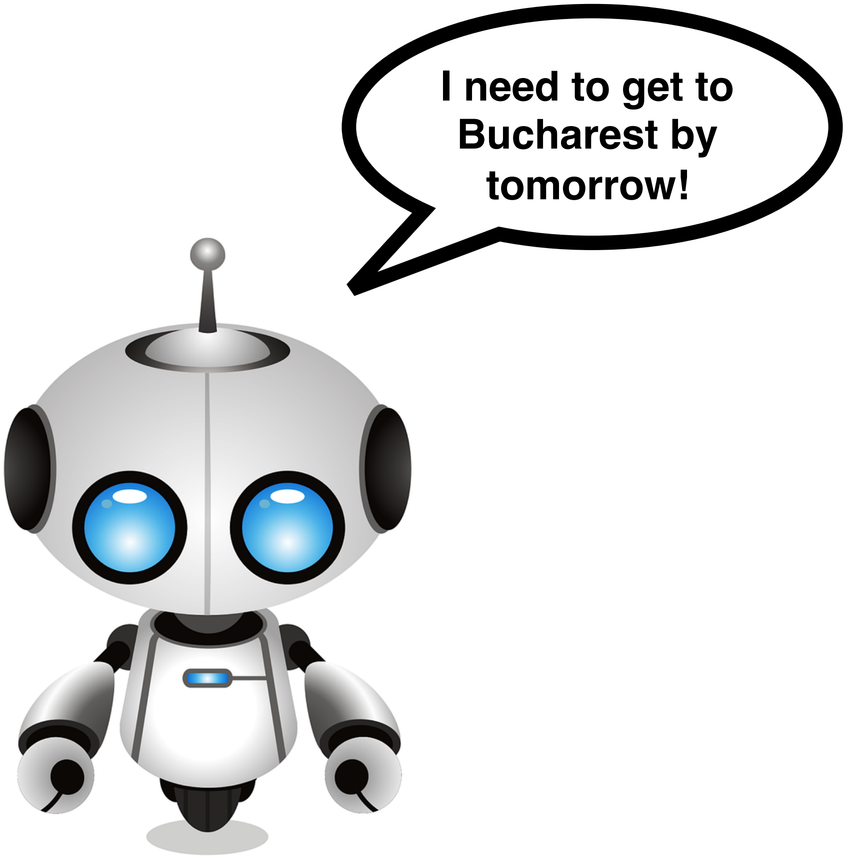
Image Credit: Russel and Norvig

Problem Solving Agents

Formulate: decide what states and actions to consider, given a goal

Search: look for a sequence of actions that reaches the goal

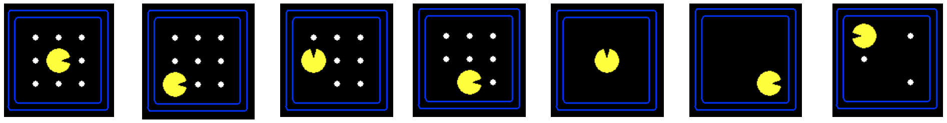
Execute: carry out the recommended actions



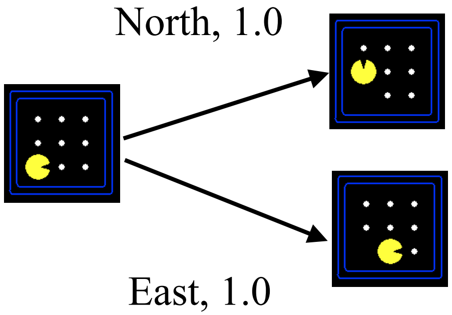
```
function SIMPLE-PROBLEM-SOLVING-AGENT( percept ) returns an action
  persistent: seq, an action sequence, initially empty
              state, some description of the current world state
              goal, a goal, initially null
              problem, a problem formulation

  state ← UPDATE-STATE( state, percept )
  if seq is empty then
      goal ← FORMULATE-GOAL( state )
      problem ← FORMULATE-PROBLEM( state, goal )
      seq ← SEARCH( problem )
      if seq = failure then return a null action
  action ← FIRST( seq )
  seq ← REST( seq )
  return action
```

What does a search problem consist of?

A state space



A successor function (with actions, costs)

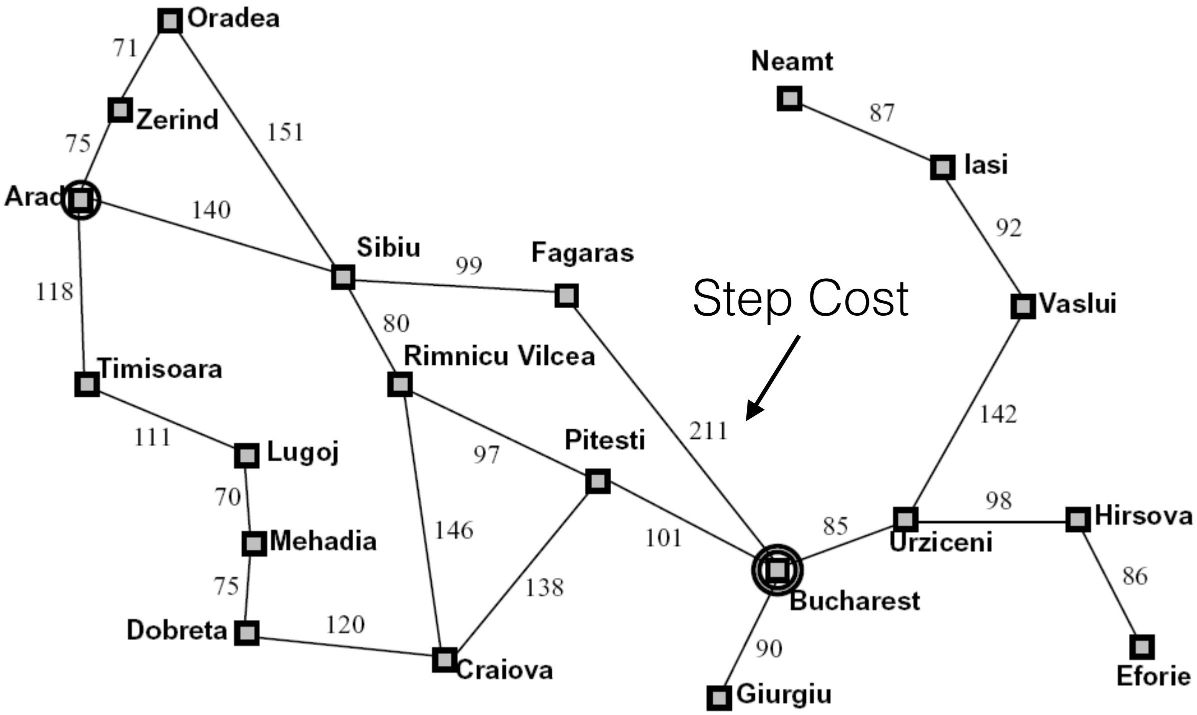


An initial state and goal test

A **solution** is a sequence of actions (a plan) that transforms the initial state to a goal state

Image credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188

Problem setup for traveling from Arad to Bucharest



Initial State: $\mathit{In}(\ \mathit{Arad}\ )$

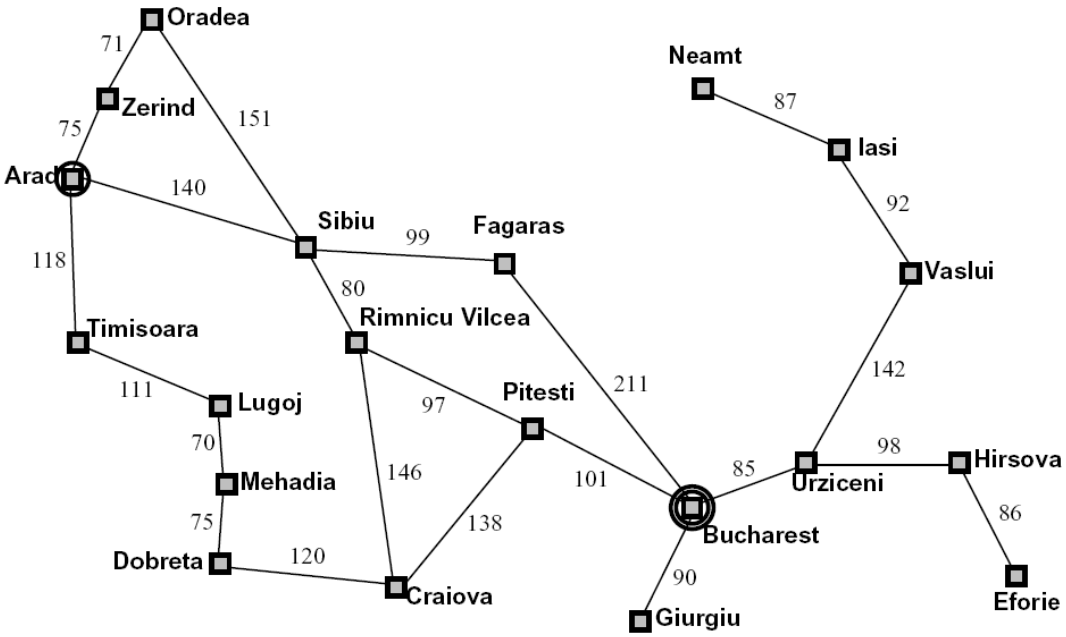
Valid Actions: $\{\mathit{Go}(\ \mathit{Sibiu}\ ), \mathit{Go}(\ \mathit{Timisoara}\ ), \mathit{Go}(\ \mathit{Zerind}\ )\}$

Successor from initial state: $\mathrm{RESULT}(\mathit{In}(\ \mathit{Arad}\ ), \mathit{Go}(\ \mathit{Zerind}\ )) = \mathit{In}(\ \mathit{Zerind}\ )$

Goal Test: $\{\mathit{In}(\ \mathit{Bucharest}\ )\}$

Path Cost: 140 (Sibiu) + 99 Fagaras + 211 Bucharest

Solution Quality



Which path is the shortest?

Optimal Solution:

Arad → Sibiu, Sibiu → Rimnicu Vilcea, Rimnicu Vilcea → Pitesti, Pitesti → Bucharest

140 + 80 + 97 + 101

Toy example: the 8-puzzle

Formulation of the 8-puzzle

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.

Actions: movements of the blank space *Up, Down, Left,* or *Right.* Different subsets of these are possible depending on where the blank is.

Formulation of the 8-puzzle

Transition model: Given a state and action, this returns the resulting state.

Goal test: This checks whether the state matches the goal configuration.

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

Computational complexity of sliding-block puzzles

The 8-puzzle is a finite problem, but it can be generalized to $n \times n$ matrices

Testing whether a solution exists is in P

Finding the solution with the fewest moves is $\mathrm{NP\text{-}complete}$

- Reduces to the 2/2/4-SAT problem

D. Ratner and M. Warmuth, *J. Symb. Comp.*, 1990

Knuth’s factorial, square root and floor sequence problem

States: Positive numbers.

Initial state: 4

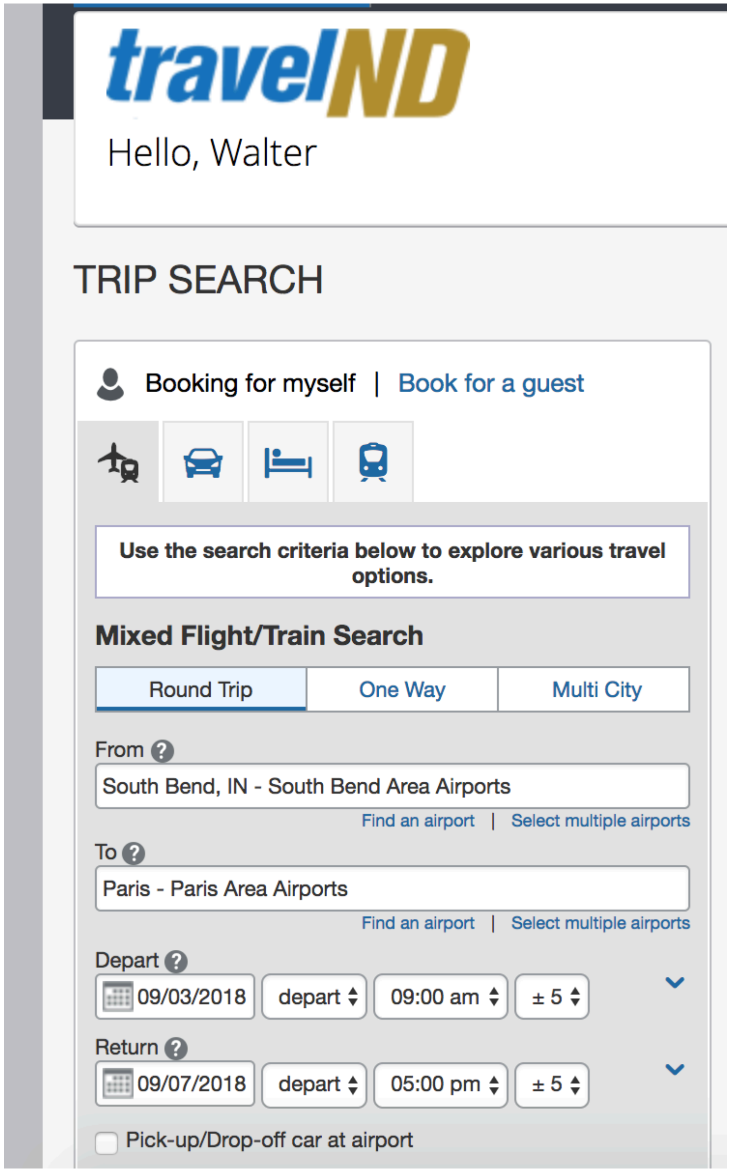
Actions: Apply factorial, square root, or floor operation

Transition Model: As given by the mathematical definitions of the operations

Goal Test: State is the desired positive integer

Real-world Problems

Route-Finding Problem



travelND

Hello, Walter

TRIP SEARCH

Booking for myself | Book for a guest

Use the search criteria below to explore various travel options.

Mixed Flight/Train Search

Round Trip | One Way | Multi City

From: South Bend, IN - South Bend Area Airports

Find an airport | Select multiple airports

To: Paris - Paris Area Airports

Find an airport | Select multiple airports

Depart: 09/03/2018 depart 09:00 am ± 5

Return: 09/07/2018 depart 05:00 pm ± 5

□ Pick-up/Drop-off car at airport

Route-Finding Problem

States: Includes a location and the current time. The state must also record extra information about a flight segment including previous segments, the fare base, and the status as domestic or international.

Initial State: This is specified by the user’s query.

Actions: Take any flight from the current location, in any seat class, leaving after the current time, leaving enough time for connecting if needed.

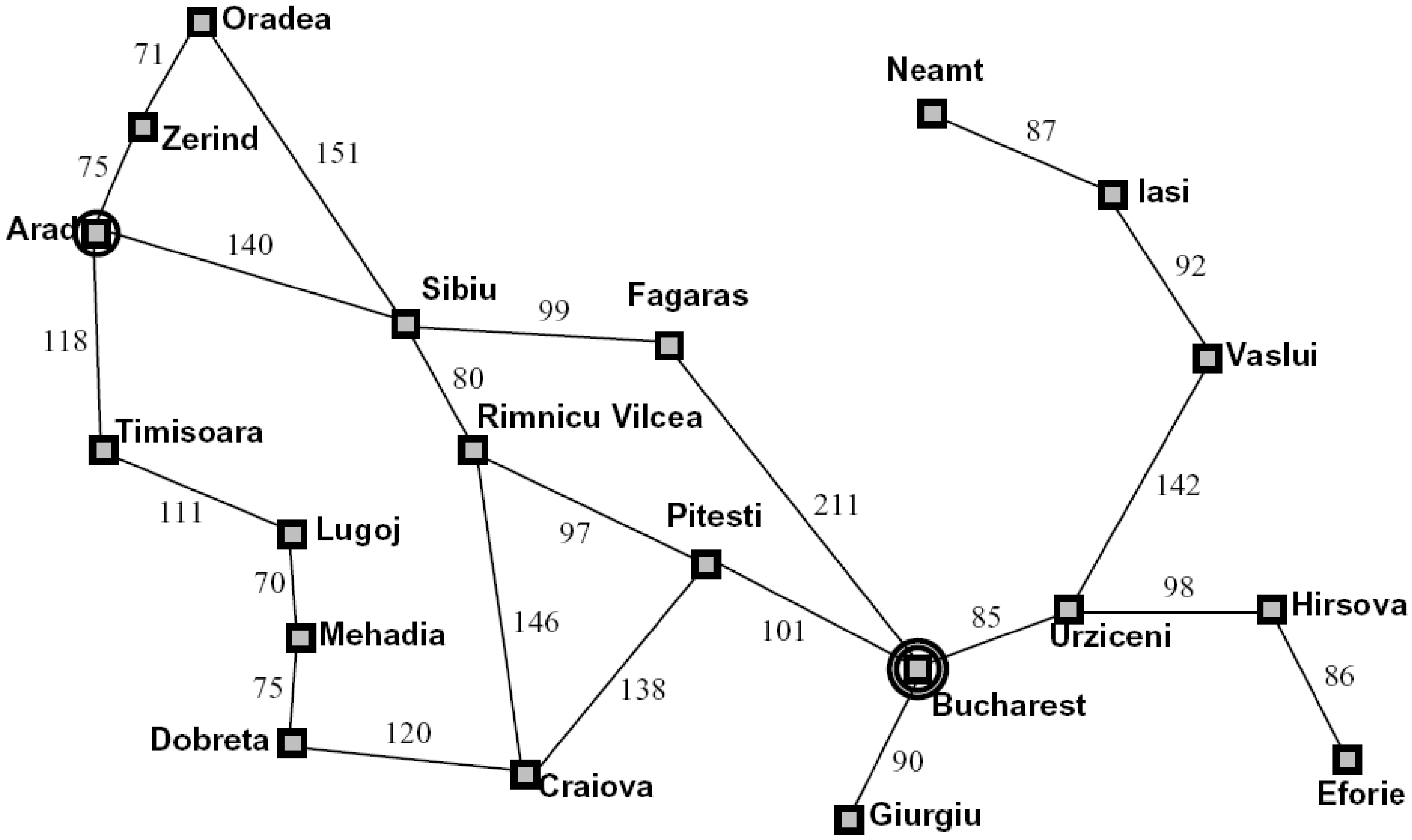
Route-Finding Problem

Transition model: The state resulting from taking a flight will have the flight's destination as the current location and the flight's arrival time as the current time.

Goal test: Are we at the final destination specified by the user?

Path cost: This depends on the monetary cost, waiting time, flight time, frequent-flyer milage awards, and so on.

Touring Problems

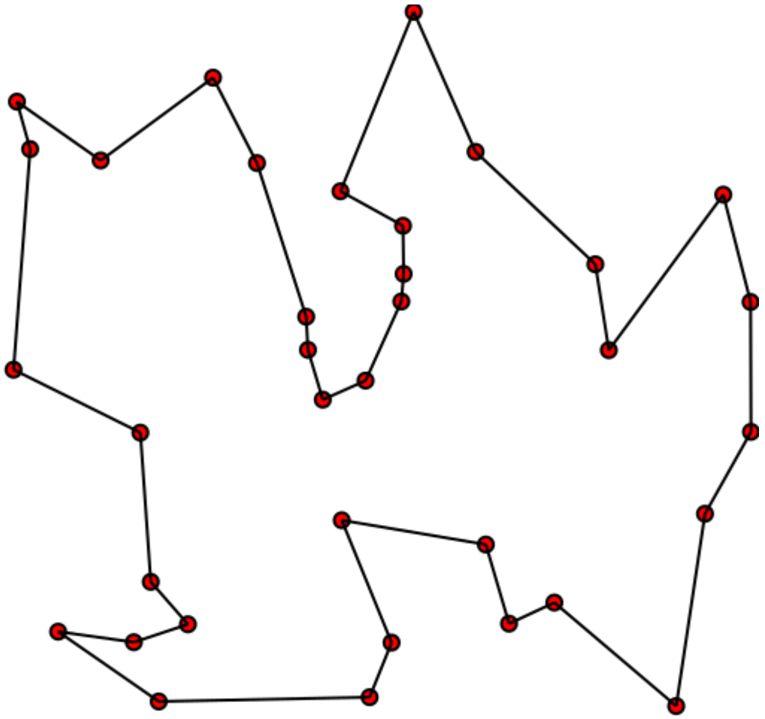


“Visit every city at least once, starting and ending in Bucharest”

Traveling Salesperson Problem

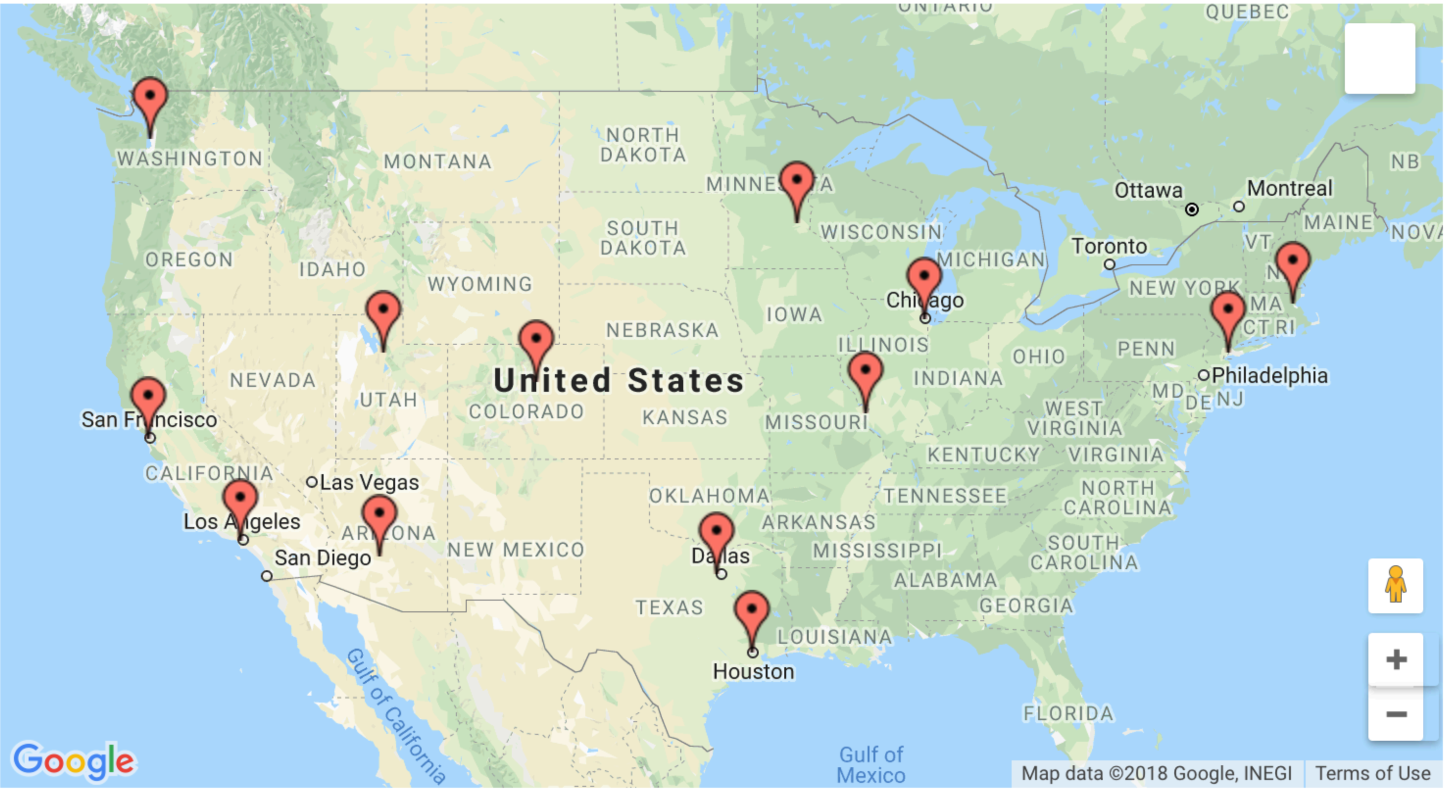
Another touring problem:

Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?



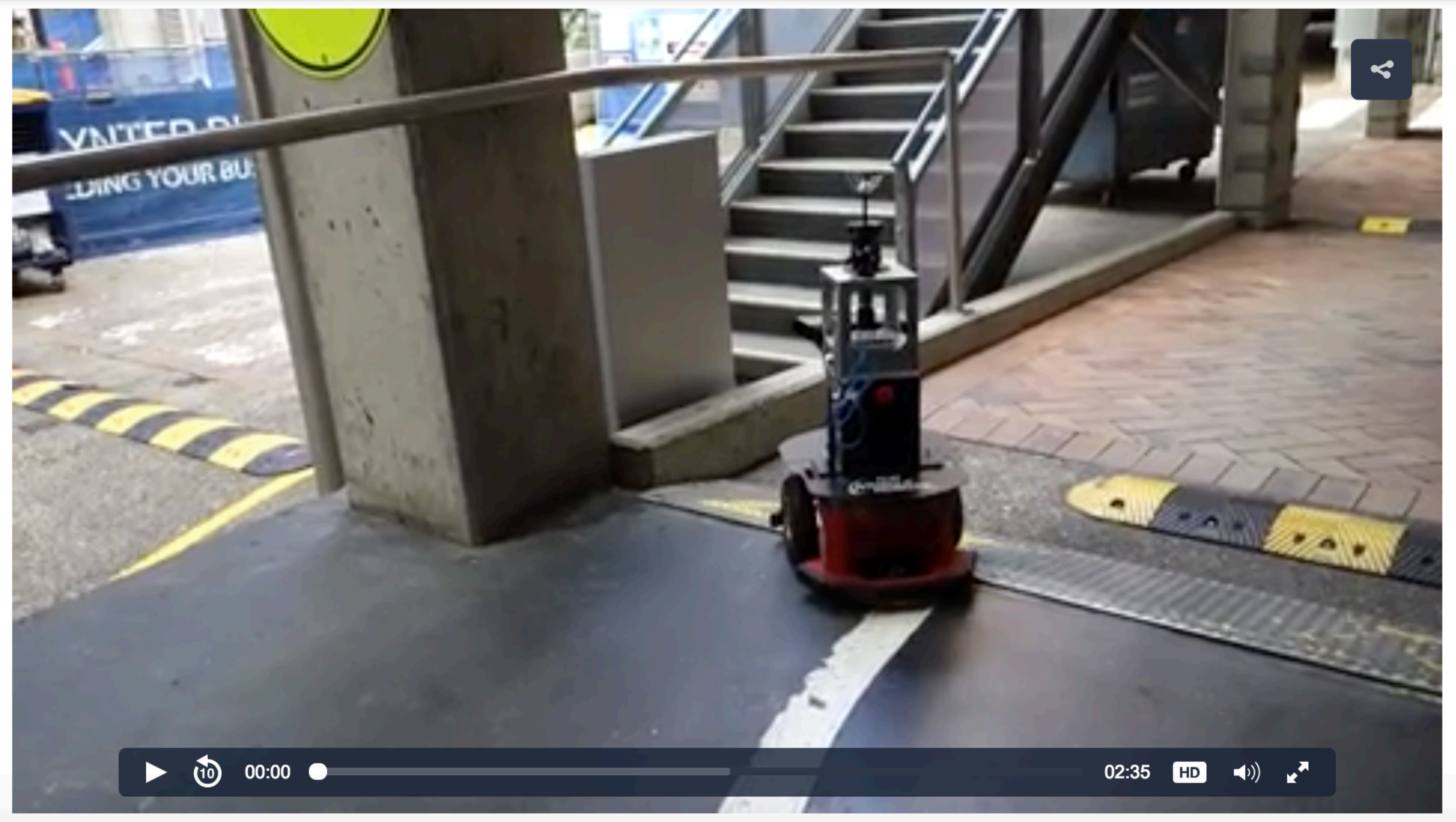
NP-hard

Google Maps and the Traveling Salesperson Problem



- Software Library: OR-Tools (https://github.com/google/or-tools/)
- Local search to improve solutions; first solutions being generated using a cheapest addition heuristic
- Optionally forbid a set of random connections between vertices

Robot Navigation



https://spectrum.ieee.org/video/robotics/robotics-software/watch-this-robot-navigate-like-a-rat

Robot Navigation

https://www.youtube.com/watch?v=aaOB-ErYq6Y