Intelligent Agents

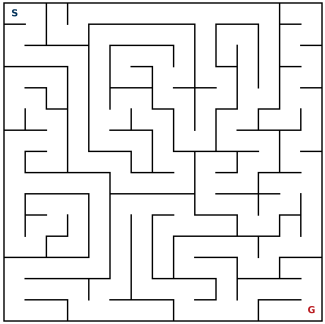
Curtis Larsen

Utah Tech University—Computing

Fall 2025

Motivation for Search

- Agents often don't know solutions in advance.
- They must **explore possible actions and consequences**.
- Search provides a systematic way to do this.

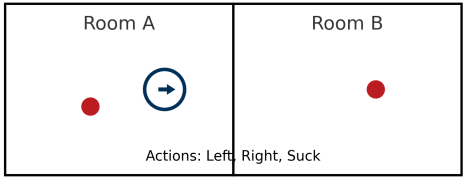


Examples of Search in Everyday AI

- Navigation (GPS directions)
- Solving puzzles (Sudoku, Rubik's cube)
- Planning tasks (robotics, scheduling)

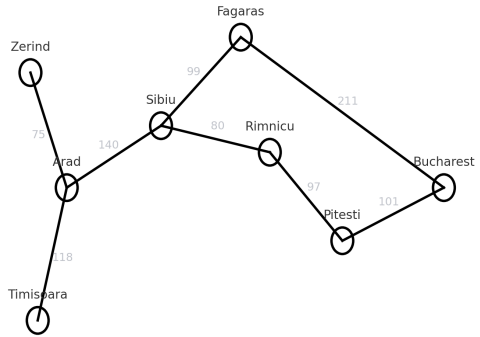
Example 1: Vacuum World (Toy Problem)

- Agent in two rooms (A and B).
- Rooms can be clean or dirty.
- Goal: both rooms clean.



Example 2: Route Finding (Medium Problem)

- States = cities (nodes).
- Actions = roads connecting cities.
- Goal: travel from start city to destination.



Example 3: Cryptarithmetic (Complex Problem)

- Puzzle: SEND + MORE = MONEY.
- States = partial digit assignments.
- Goal: complete valid assignment.

```
  S E N D
+ M O R E
---------
M O N E Y
```

Letter-to-Digit Mapping

S	E	N	D	M	O	R	Y
•	•	•	•	•	•	•	•

Defining a State

- A **state** = description of the current situation.
- Should capture all relevant information to decide what to do next.

Think-Pair-Share: How would you define a state in:

- Vacuum World?
- Route Finding?
- Cryptarithmetic?

Initial and Goal States

- **Initial state**: where the agent starts.
- **Goal test**: condition to check for success.

Prompt: What are the initial and goal states in each example?

Actions and Successor Functions

- **Actions**: available choices at a state.
- **Successor function**: mapping from state + action $\rightarrow$ new state.

Prompt: List actions in Vacuum World, Route Finding, Cryptarithmetic.

Transition Models

- **Transition model**: describes what happens when an action is taken.
- Can be deterministic (predictable) or nondeterministic (uncertain).

Example: driving between cities (sometimes nondeterministic: traffic, weather).

Path Costs

- **Path cost**: numerical value for a sequence of actions.
- Defines solution quality (shortest, cheapest, fastest).

Prompt: What is a natural path cost in each of our three examples?

Problem Formulation Recap

A search problem is defined by 5 components:

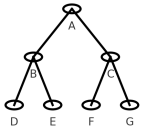
1. **Initial state:** s_0
   (the starting point of the search)
2. **Actions:** $A(s) \rightarrow \{a_1, a_2, \dots\}$
   Returns the set of possible actions in state s
3. **Transition model:** $T(s, a) \rightarrow s'$
   Returns the resulting state when action a is applied in state s
4. **Goal test:** $G(s) \rightarrow \{\text{true}, \text{false}\}$
   Checks whether state s is a goal state
5. **Path cost:** $C(s, a, s') \rightarrow \mathbb{R}_{\geq 0}$
   Assigns a numeric cost to the step from s to s' via a

Prompt: How do these functions look in our 3 examples?

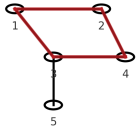
Search Trees vs. Graphs

- **Tree search**: may revisit states repeatedly.
- **Graph search**: avoids repeated states.
- Important for efficiency and correctness.

Tree Search



Graph Search



Measuring Search Performance

Evaluation criteria:

- **Completeness**: guaranteed to find solution?
- **Optimality**: guaranteed to find best solution?
- **Time complexity**: how long?
- **Space complexity**: how much memory?

Day 1 Wrap-Up

- Problem formulation = defining states, actions, goals, costs.
- Performance measured by completeness, optimality, time, space.
- Next time: algorithms that actually search.