Search

Introduction to Artificial Intelligence

COS302

Michael L. Littman

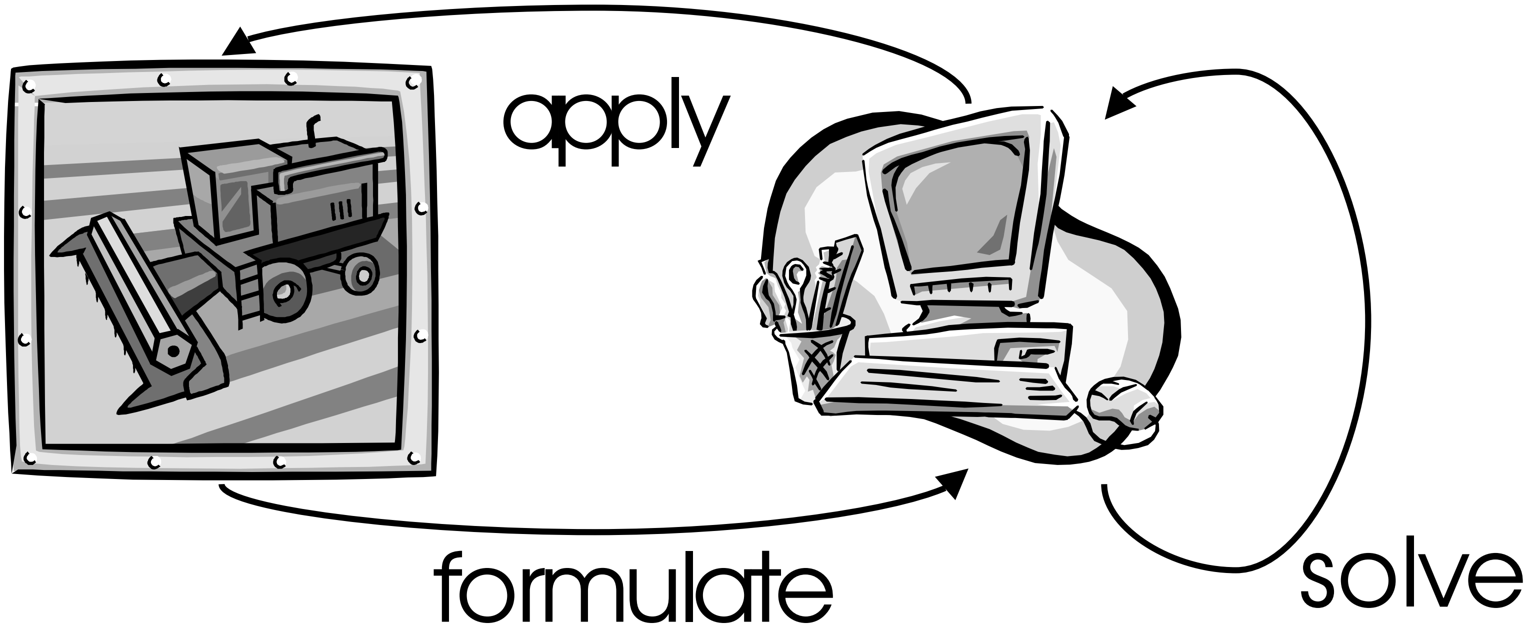
Fall 2001

Administration

- Short written homeworks each week
- First one today
- Web page is up with first lecture
- Send me (mlittman@cs) your email address so I can make a mailing list
- Office hours...

What’s AI? (to me)

Computers making decisions in real-world problems



apply

formulate

solve

Search Problems

Let $\mathcal{S}$ be the set of states (strings)

Input:

- Initial state: s_0
- Neighbor generator, N: $\mathcal{S} \rightarrow 2^{\mathcal{S}}$
- Goal function, G: $\mathcal{S} \rightarrow \{0,1\}$

Search Answer

$s_1, \ldots, s_n$ such that:

- $s_1, \ldots, s_n \in \mathcal{S}$
- for all $1 \leq i \leq n$, $s_i \in N(s_{i-1})$
- $G(s_n) = 1$

Examples

We’re very impressed. Meaning?

- **Rush Hour**
- **8-puzzle**
- **Logistics**
- **8-queens problem**
- **Logic puzzles**
- **Job-shop scheduling**

Rush Hour

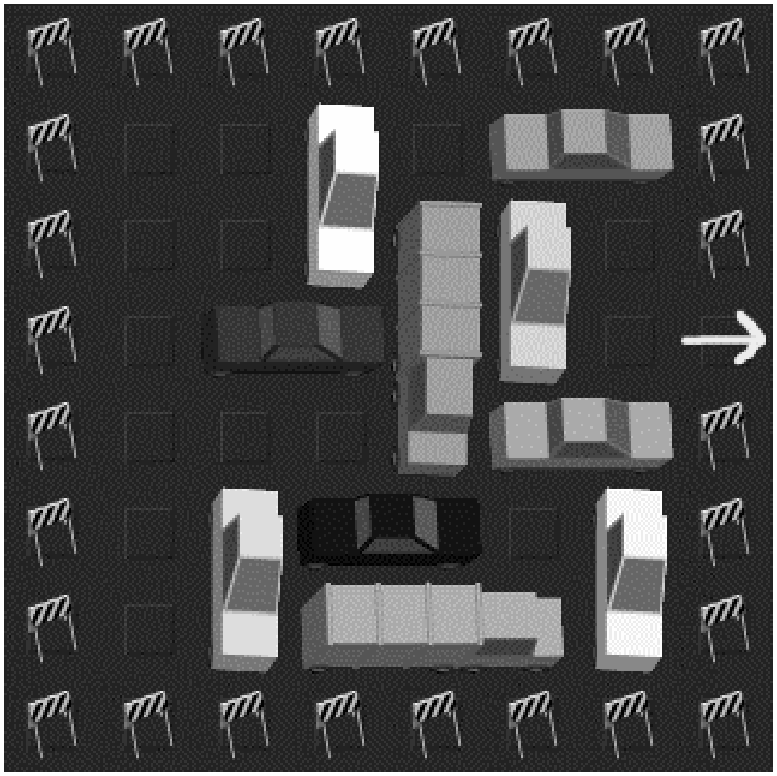
Move cars forward and backward to “escape”

Search Version

States: configurations of cars

N(s): reachable states

G(s): 1 if red car at gate



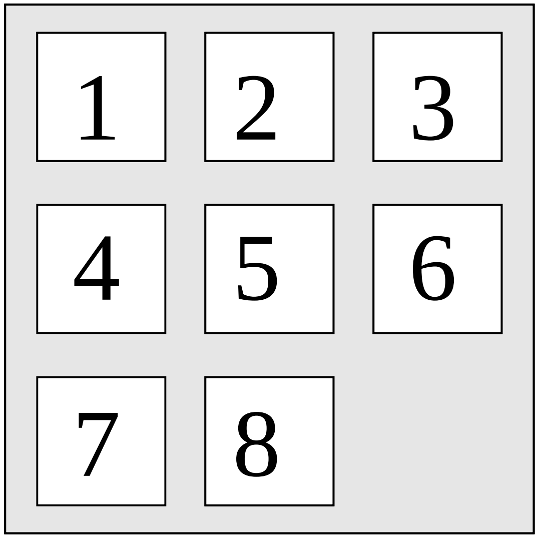
8-puzzle

Slide tiles into order

States:

N(s):

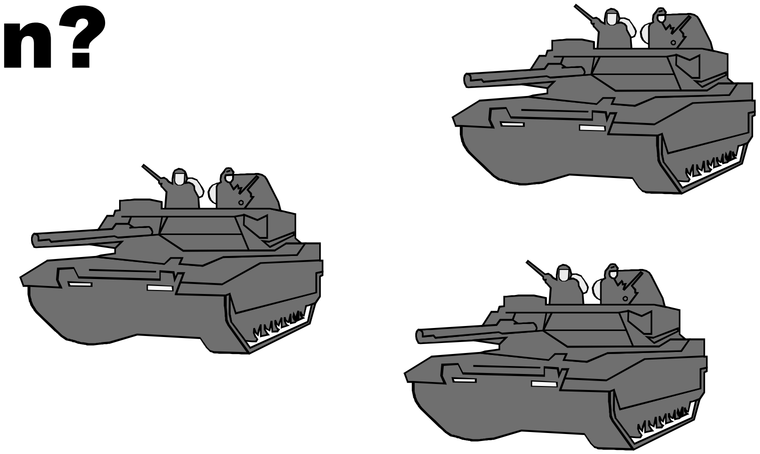
G(s):



1	2	3
4	5	6
7	8	

Logistics

Very sophisticated. What goes where when?



Desert Storm logistics “paid for AI research”

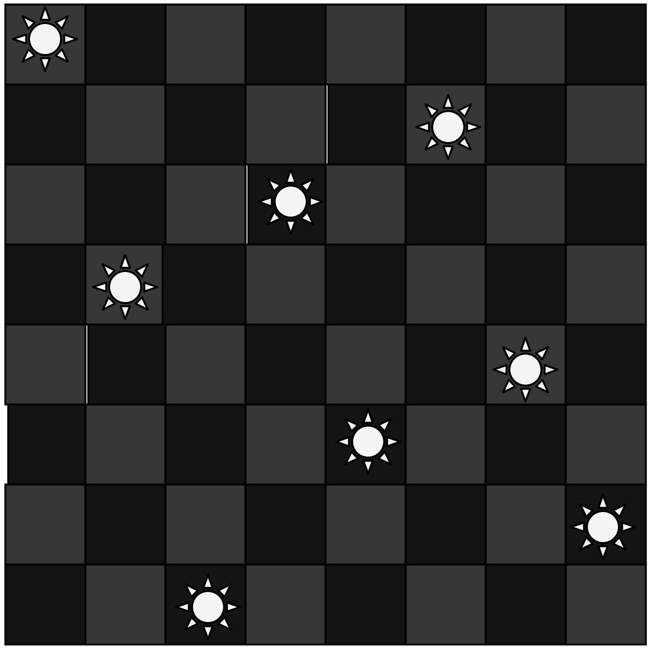
8 Queens Puzzle

No captures

States:

N(s):

G(s):



Logic Puzzles

1. Jody, who is an ape, wasn’t the ape who returned immediately after Tom and immediately before the animal who appeared in the movie with no rating.
2. The only lions that were used in the movies were the one who was the third to return, the one who appeared in the R movie, and the one who appeared in “Luck”. …

Job-Shop Scheduling

Industrial problem:

- Allocate machines and machinists to time slots
- Constraints on orders in which parts are serviced

Search Template

```
• fringe = {($s_0$, 0)};   /* initial cost */
• markvisited($s_0$);
• While (1) {
   If empty(fringe), return failure;
   (s, c) = removemincost(fringe);
   If G(s) return s;
   Foreach s’ in N(s)
      if unvisited(s’)
        fringe = fringe U {(s’, cost(s’)};
      markvisited($s_0$);
  }
```

Data Structures

How implement this efficiently?

- **removemincost-U-empty?**
- **markvisited-unvisited?**

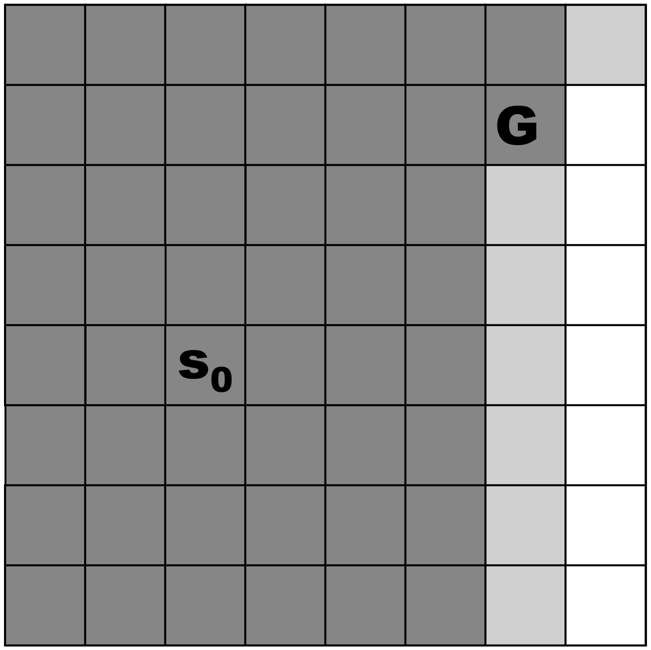
Vary Cost

How does search behavior change with cost?

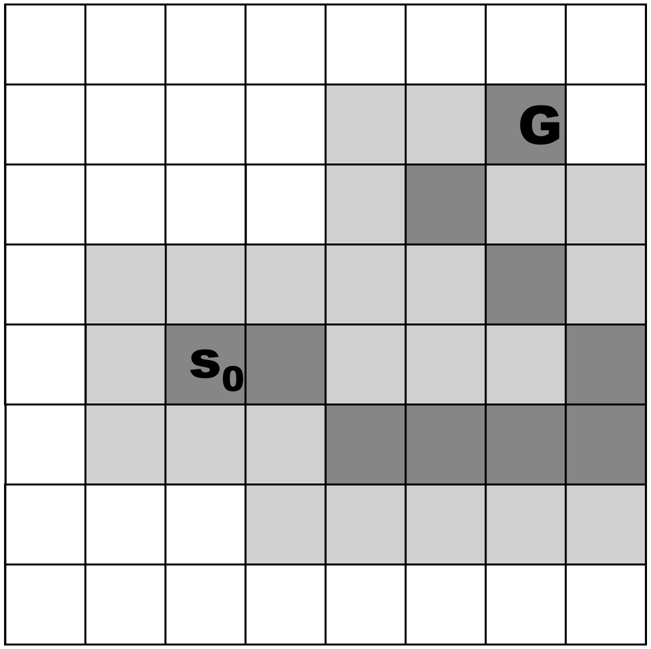
- cost(s’) = c + 1

- cost(s’) = c - 1

Grid Example: BFS



Grid Example: DFS



How Evaluate?

What makes one search scheme better than another?

- Completeness: Find solution?
- Time complexity: How long?
- Space complexity: Memory?
- Optimality: Find shortest path?

Depth vs. Breadth-first

Let $|T(s)| \leq b$ (branching factor), goal at depth d

- How implement priority queue?
- Completeness?
- Time complexity?
- Space complexity?
- Optimality?

BFS

- Completeness?
  - – Yes
- Time complexity?
  - – $O(b^d)$
- Space complexity?
  - – $O(b^d)$ ☹
- Optimality?
  - – yes

DFS

- Completeness?
  - Yes, assuming state space finite
- Time complexity?
  - O(|$\mathcal{S}$ |), can do well if lots of goals
- Space complexity?
  - O(n), n deepest point of search
- Optimality?
  - No ☹

Depth-limited Search

DFS, only expand nodes depth ≤ l.

- **Completeness?**
  - **No, if l ≤ d. ☹**
- **Time complexity?**
  - **O(b^l)**
- **Space complexity?**
  - **O(l)**
- **Optimality?**
  - **No ☹**

Iterative Deepening

Depth limited, increasing l.

- **Completeness?**
  - **Yes. ☺**
- **Time complexity?**
  - **$O(b^d)$, even with repeated work! ☺**
- **Space complexity?**
  - **$O(d)$ ☺**
- **Optimality?**
  - **Yes ☺**

Bidirectional Search

BFS in both directions

Need N^{-1}

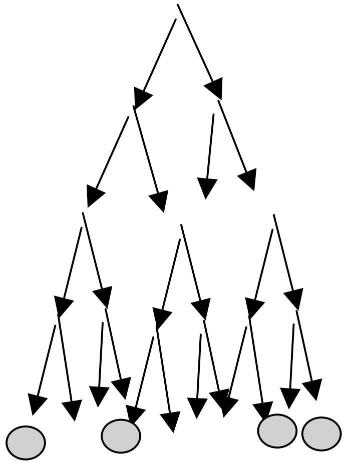
How could this help?

- b^l vs $2b^{l/2}$

What makes this hard to implement?

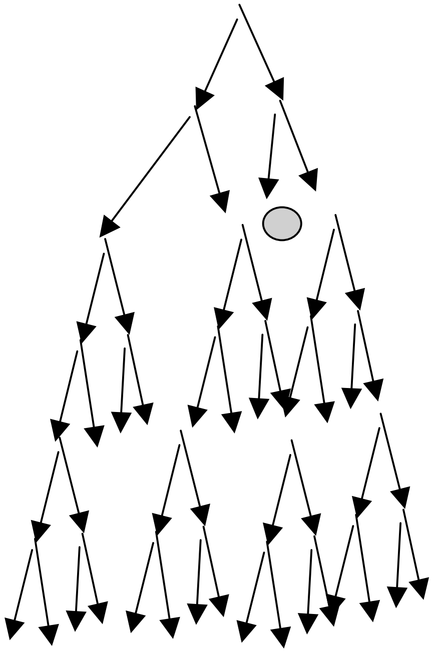
Which do you choose?

- 8-queens, neighbors of s add one queen to board



Which do you choose?

- Big grid, goal nearby



What to Learn

- How to express problems in the search framework
- The basic algorithms for search
- Strengths and weaknesses of the basic algorithms

Homework 1 (due 9/26)

1. Send your email address (right away) to littman@cs.
2. Let BFS' and DFS' be versions of BFS and DFS that don't check whether a state has been previously visited. Evaluate BFS' and DFS' on the four comparison criteria we discussed.
3. Consider the Rush Hour board from these notes. Assume a single move consists of sliding a car forward or backward some number of spaces. (a) Give an upper bound on the branching factor. (b) Assuming the solution is at depth 20, how many nodes will be searched? (c) Ignoring the search, how many states are in the search space? Give as tight an upper bound as you can.