19: Informed search
Lecture 19: Informed search

- Foxes and chickens
- Search problems
- Informed search
Three foxes and three chickens wish to cross the river. They have a small boat that will carry up to two animals. Everyone can navigate the boat. If at any time the foxes outnumber the chickens on either bank of the river, they will eat the chickens. Find the smallest number of crossings that will allow everyone to cross the river safely.

What is the “state” of this problem (it should capture all possible valid configurations)?
Foxes and Chickens

Three foxes and three chickens wish to cross the river. They have a small boat that will carry up to two animals. Everyone can navigate the boat. If at any time the foxes outnumber the chickens on either bank of the river, they will eat the chickens. Find the smallest number of crossings that will allow everyone to cross the river safely.
Foxes and Chickens

Three foxes and three chickens wish to cross the river. They have a small boat that will carry up to two animals. Everyone can navigate the boat. If at any time the foxes outnumber the chickens on either bank of the river, they will eat the chickens. Find the smallest number of crossings that will allow everyone to cross the river safely.

```
FFFCCC B
FFCC  B FC
FC   B FFCC
...
```
Searching for a solution

What states can we get to from here?
Searching for a solution

FOXES AND CHICKENS

Next states?
Fox and Chickens Solution

How is this solution different than the n-queens problem?
Solution is not a state, but a sequence of actions (or a sequence of states)
Look at the code that solves the foxes and chickens problem.
One other problem

What would happen if we ran DFS here?
FOXES AND CHICKENS

One other problem

If we always go left first, will continue forever!

If we always go left first, will continue forever!
One other problem

Does BFS have this problem?
One other problem

No!
DFS vs. BFS

- Why do we use DFS then, and not BFS?
Consider a search problem where each state has two states you can reach.

Assume the goal state involves 20 actions, i.e. moving between ~20 states.

How big can the queue get for BFS?
Consider a search problem where each state has two states you can reach.

Assume the goal state involves 20 actions, i.e. moving between ~20 states.

At any point, need to remember roughly a level.
DFS vs. BFS

Consider a search problem where each state has two states you can reach.

Assume the goal state involves 20 actions, i.e. moving between ~20 states.

How big does this get?
Consider a search problem where each state has two states you can reach.

Assume the goal state involves 20 actions, i.e. moving between ~20 states.

Doubles every level we have to go deeper. For 20 actions that is $2^{20} \approx 1$ million states!
Consider a search problem where each state has two states you can reach.

Assume the goal state involves 20 actions, i.e. moving between ~20 states.

How many states would DFS keep on the stack?
Consider a search problem where each state has two states you can reach.

Assume the goal state involves 20 actions, i.e. moving between ~20 states.

Only one path through the tree, roughly 20 states.
One other problem

If we always go left first, will continue forever!

Solution?
DFS avoiding repeats

```python
def dfs(state, visited):
    # note that we've visited this state
    visited[str(state)] = True

    if state.is_goal():
        return [state]
    else:
        result = []

        for s in state.next_states():
            # check if we've visited a state already
            if not str(s) in visited:
                result += dfs(s, visited)

        return result
```
Lecture 19: Informed search

- Foxes and chickens
- Search problems
- Informed search
Other search problems

What problems have you seen that could be posed as search problems by defining:

- What is the state
- Start state
- Goal state
- State-space/transition between states
8-puzzle

Start State

Goal State
8-puzzle

- goal
  
  ![Goal State]

- state representation?

- start state?

- state-space/transitions?
8-puzzle

state:
- all 3 x 3 configurations of the tiles on the board

transitions between states:
- Move Blank Square Left, Right, Up or Down.
- This is a more efficient encoding than moving each of the 8 distinct tiles
8-puzzle
Cryptarithmetic

Find an assignment of digits (0, ..., 9) to letters so that a given arithmetic expression is true. examples:

\[
\text{SEND} + \text{MORE} = \text{MONEY}
\]

\[
\begin{array}{c}
\text{FORTY} \\
+ \text{TEN} \\
+ \text{ TEN} \\
+ \text{ TEN} \\
\hline
\text{SIXTY}
\end{array}
\]

\[F=2, \ O=9, \ R=7, \ etc.\]
Cryptarithmetic

Find an assignment of digits (0, ..., 9) to letters so that a given arithmetic expression is true. examples:

\[
\begin{align*}
\text{SEND} + \text{MORE} &= \text{MONEY} \\
\text{FORTY} &\quad \text{Solution:} \quad 29786 \\
+ \text{TEN} &\quad \quad 850 \\
+ \text{TEN} &\quad \quad 850 \\
\hline \\
\text{SIXTY} &\quad \quad 31486 \\
\end{align*}
\]

\(F=2, \ O=9, \ R=7, \) etc.
Remove 3 match sticks

Given the following configuration of sticks, remove exactly 3 sticks in such a way that the remaining configuration forms exactly 3 squares.
Remove 3 match sticks

Given the following configuration of sticks, remove exactly 3 sticks in such a way that the remaining configuration forms exactly 3 squares.
Remove 5 match sticks

Given the following configuration of sticks, remove exactly 5 sticks in such a way that the remaining configuration forms exactly 3 squares.
Remove 5 match sticks

Given the following configuration of sticks, remove exactly 5 sticks in such a way that the remaining configuration forms exactly 3 squares.
Water Jug Problem

- Given a full 5-gallon jug and a full 2-gallon jug, fill the 2-gallon jug with exactly one gallon of water.
## SEARCH PROBLEMS

<table>
<thead>
<tr>
<th>Name</th>
<th>Cond.</th>
<th>Transition</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty5</td>
<td>-</td>
<td>((x, y) \rightarrow (0, y))</td>
<td>Empty 5-gal. jug</td>
</tr>
<tr>
<td>Empty2</td>
<td>-</td>
<td>((x, y) \rightarrow (x, 0))</td>
<td>Empty 2-gal. jug</td>
</tr>
<tr>
<td>2to5</td>
<td>(x \leq 3)</td>
<td>((x, 2) \rightarrow (x+2, 0))</td>
<td>Pour 2-gal. into 5-gal.</td>
</tr>
<tr>
<td>5to2</td>
<td>(x \geq 2)</td>
<td>((x, 0) \rightarrow (x-2, 2))</td>
<td>Pour 5-gal. into 2-gal.</td>
</tr>
<tr>
<td>5to2part</td>
<td>(y &lt; 2)</td>
<td>((1, y) \rightarrow (0, y+1))</td>
<td>Pour partial 5-gal. into 2-gal.</td>
</tr>
</tbody>
</table>
8-puzzle revisited

How hard is this problem?

```
1 3 8
4 7
6 5 2
```
8-puzzle revisited

- The average depth of a solution for an 8-puzzle is 22 moves
- An exhaustive search requires searching $\sim 3^{22}$ states
  - BFS: 10 terabytes of memory
  - DFS: 8 hours (assuming one million nodes/second)
- Can we do better?
- Is DFS and BFS intelligent?
Lecture 19: Informed search

- Foxes and chickens
- Search problems
- Informed search
How do you think Google Maps does it?
What would the search algorithms do?
INFORMED SEARCH

DFS
INFORMED SEARCH

Ideas?
We’d like to bias search towards the actual solution
INFORMED SEARCH

Informed search

- Order to_visit based on some knowledge of the world that estimates how “good” a state is
  - h(n) is called an evaluation function

- Best-first search
  - rank to_visit based on h(n)
  - take the most desirable state in to_visit first
  - different approaches depending on how we define h(n)
Heuristic

- Merriam-Webster's Online Dictionary
  - Heuristic (pron. \hyu-\-'ris-tik\): adj. [from Greek heuriskein to discover.] involving or serving as an aid to learning, discovery, or problem-solving by experimental and especially trial-and-error methods

- The Free On-line Dictionary of Computing (2/19/13)
  - heuristic 1. Of or relating to a usually speculative formulation serving as a guide in the investigation or solution of a problem: "The historian discovers the past by the judicious use of such a heuristic device as the 'ideal type'" (Karl J. Weintraub).
Heuristic function: $h(n)$

- An estimate of how close the node is to a goal
- Uses domain-specific knowledge!

Examples

- **Map path finding?**
  - straight-line distance from the node to the goal ("as the crow flies")

- **8-puzzle?**
  - how many tiles are out of place
  - sum of the "distances" of the out of place tiles

- **Foxes and Chickens?**
  - number of passengers on the starting bank
TODAY’S LECTURE IN A NUTSHELL

Lecture 19: Informed search

- Foxes and chickens
- Search problems
- Informed search
Resources

- chickens.py

Homework

- Assignment 9 (cont’d)