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.

```
  FFFCCC B
  FFCC     B FC
  FC       B FFCC
  ...
```

Searching for a solution

```
FFFCCC B ~
```

What states can we get to from here?

Searching for a solution

```
Next states?
```

Fox and Chickens Solution

```
How is this solution different than the n-queens problem?
```
Fox and Chickens Solution

Solution is not a state, but a sequence of actions (or a sequence of states)

One other problem

What would happen if we ran DFS here?

One other problem

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

Does BFS have this problem?

No!

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 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 “row”.

How big does this get?

Doubles every level we have to go deeper. For 20 actions that is $2^{20} \approx 1$ million states!

How many states would DFS keep on the stack?

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. Only one path through the tree, roughly 20 states.

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[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(s in visited):
                result += dfs(s, visited)
        return result
```

Other search problems

What problems have you seen that could be posed as search problems?

What is the state?

Start state

Goal state

State-space/transition between states
8-puzzle

goal

state representation?

start state?

state-space/transitions?

---

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
Cryptarithmetic

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

examples:

SEND + MORE = MONEY

FORTY
+ TEN
+ TEN
-----
SIXTY
F=2, O=9, R=7, etc.

Remove 5 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.

State = (x,y), where x is the number of gallons of water in the 5-gallon jug and y is # of gallons in the 2-gallon jug
Initial State = (5,2)
Goal State = (*,1), where * means any amount

Water Jug Problem

Operator table

<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)→(0,y)</td>
<td>Empty 5-gal. jug</td>
</tr>
<tr>
<td>Empty2</td>
<td>-</td>
<td>(x,y)→(x,0)</td>
<td>Empty 2-gal. jug</td>
</tr>
<tr>
<td>2x5</td>
<td>x ≤ 3</td>
<td>(x,2)→(x+2,0)</td>
<td>Pour 2-gal. into 5-gal.</td>
</tr>
<tr>
<td>5x2</td>
<td>x ≥ 2</td>
<td>(x,0)→(x,2,0)</td>
<td>Pour 5-gal. into 2-gal.</td>
</tr>
<tr>
<td>5x2part</td>
<td>y &lt; 2</td>
<td>(1,y)→(0,y+1)</td>
<td>Pour partial 5-gal. into 2-gal.</td>
</tr>
</tbody>
</table>
8-puzzle revisited

How hard is this problem?

The average depth of a solution for an 8-puzzle is 22 moves

An exhaustive search requires searching $\sim 3^{22} \approx 3.1 \times 10^{10}$ states

- BFS: 10 terabytes of memory
- DFS: 8 hours (assuming one million nodes/second)

Can we do better?

Is DFS and BFS intelligent?

from: Claremont to: Rowland Heights

How do you think google maps does it?

What would the search algorithms do?
5/11/22

from: Claremont to: Rowland Heights

DFS

from: Claremont to: Rowland Heights

BFS

37

38

from: Claremont to: Rowland Heights

Ideas?

from: Claremont to: Rowland Heights

We’d like to bias search towards the actual solution

39

40
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 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 people on the starting bank

Heuristic

Merriam-Webster’s Online Dictionary

Heuristic (pron. \ˈhī-ər-ik\): 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).