

**INFORMED SEARCH**

David Kauchak  
CS51A – Spring 2022

1

Admin

---

Assignment 9

2

**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.

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

3

**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.

4

## 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.

FFFCC B ~

FFCC            B FC

FC                B FFCC

...

5

## Searching for a solution

FFFCC B ~

What states can we get to from here?

6

## Searching for a solution

FFFCC B ~

FFCC ~ B F

FFCC ~ B FC

FFCC ~ B FF

Next states?

7

## Fox and Chickens Solution

```

FFFCC B|~~~~~|
FFCC  |~~~~~|B FC
FFCC B|~~~~~| F
CCC   |~~~~~|B FFF
FFCC B|~~~~~| FF
FC    |~~~~~|B FFCC
FFCC B|~~~~~| FC
FF    |~~~~~|B FFCC
FFF B |~~~~~| CCC
F     |~~~~~|B FFCC
FC   B|~~~~~| FFCC
     |~~~~~|B FFCC
    
```

How is this solution different than the n-queens problem?

8

### Fox and Chickens Solution

```

FFFCCC B|~~~~~|
FFCC  |~~~~~|B FC
FFCCC B|~~~~~| F
CCC   |~~~~~|B FFF
FCCC B|~~~~~| FF
FC    |~~~~~|B FFCC
FFCC B|~~~~~| FC
FF    |~~~~~|B FCCC
FFF  B|~~~~~| CCC
F    |~~~~~|B FFCCC
FC  B|~~~~~| FFCC
    |~~~~~|B FFFCCC
    
```

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

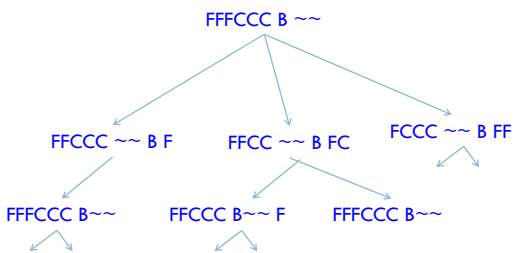
9

### Code!

<https://cs.pomona.edu/classes/cs51a/examples/chickens.txt>

10

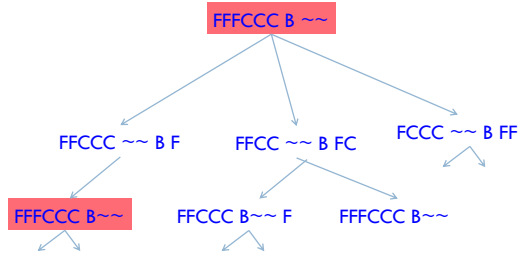
### One other problem



What would happen if we ran DFS here?

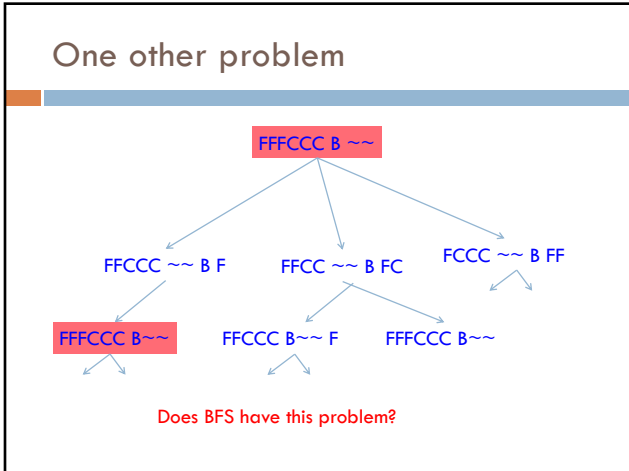
11

### One other problem

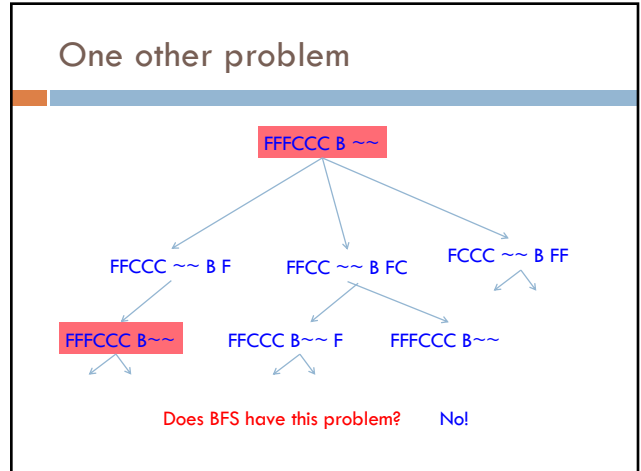


If we always go left first, will continue forever!

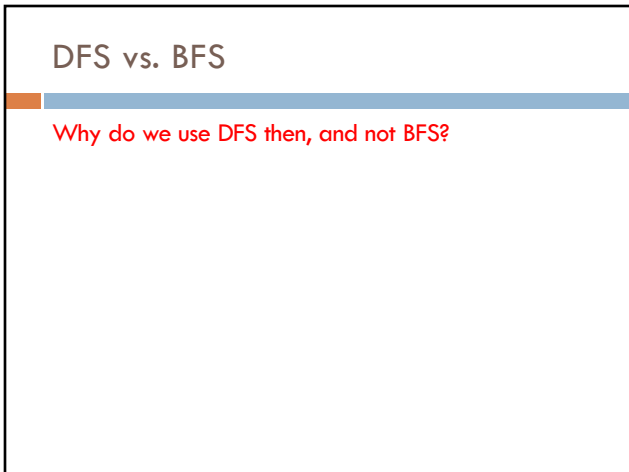
12



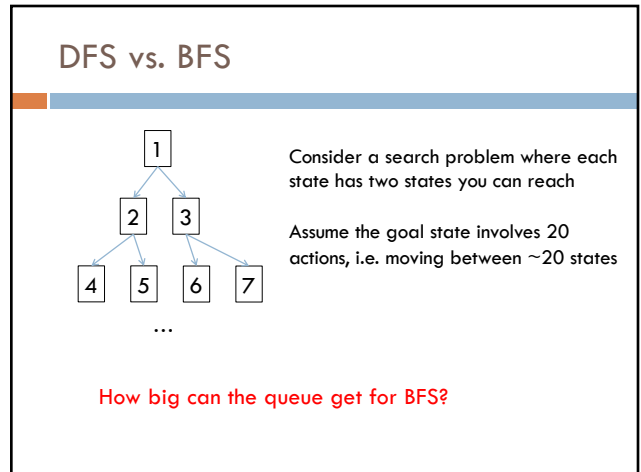
13



14



15



16

### 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

At any point, need to remember roughly a "row"

17

### 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?

18

### 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

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

19

### DFS vs. BFS

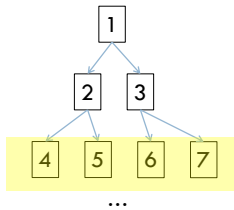
Consider a search problem where each state has two states you can reach

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

How many states would DFS keep on the stack?

20

## DFS vs. BFS



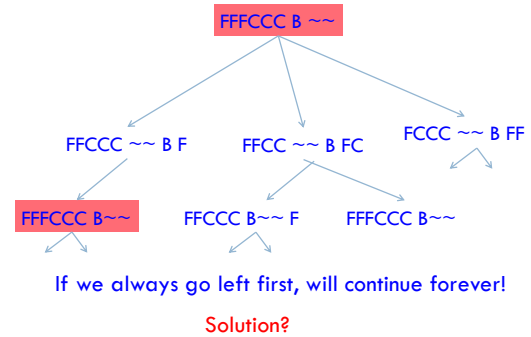
Consider a search problem where each state has two states you can reach

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

Only one path through the tree, roughly 20 states

21

## One other problem



22

## DFS avoiding repeats

```

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
  
```

23

## 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

24

### 8-puzzle

5	4	
6	1	8
7	3	2

1	2	3
8		4
7	6	5

Start State
Goal State

25

### 8-puzzle

goal

1	2	3
8		4
7	6	5

Goal State

state representation?

start state?

state-space/transitions?

26

### 8-puzzle

**state:**

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

**transitions between states:**

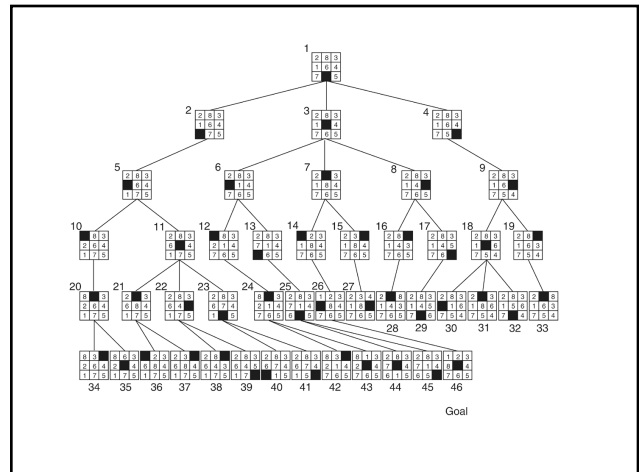
- Move Blank Square Left, Right, Up or Down.
- This is a more efficient encoding than moving each of the 8 distinct tiles

5	4	
6	1	8
7	3	2

1	2	3
8		4
7	6	5

Start State
Goal State

27



28

### Cryptarithmic

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

$$\text{SEND} + \text{MORE} = \text{MONEY}$$

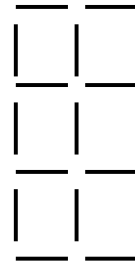
```

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

29

### 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.



30

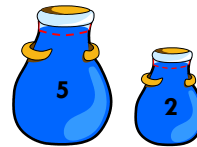
### Water Jug Problem

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



31

### Water Jug Problem



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

Operator table

Name	Cond.	Transition	Effect
Empty5	-	(x,y)→(0,y)	Empty 5-gal. jug
Empty2	-	(x,y)→(x,0)	Empty 2-gal. jug
2to5	$x \leq 3$	$(x,2) \rightarrow (x+2,0)$	Pour 2-gal. into 5-gal.
5to2	$x \geq 2$	$(x,0) \rightarrow (x-2,2)$	Pour 5-gal. into 2-gal.
5to2part	$y < 2$	$(1,y) \rightarrow (0,y+1)$	Pour partial 5-gal. into 2-gal.

32



## 8-puzzle revisited

How hard is this problem?

1	3	8
4		7
6	5	2

33

## 8-puzzle revisited

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

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

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

Can we do better?

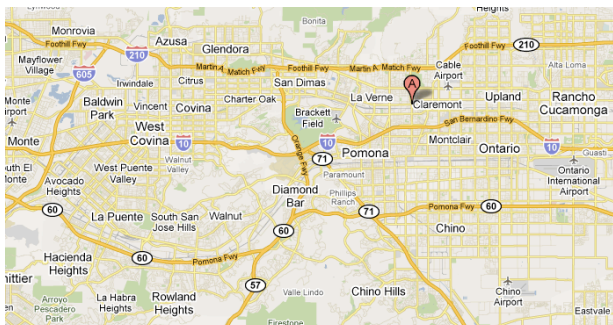
Is DFS and BFS intelligent?

1	3	8
4		7
6	5	2

34

## from: Claremont to: Rowland Heights

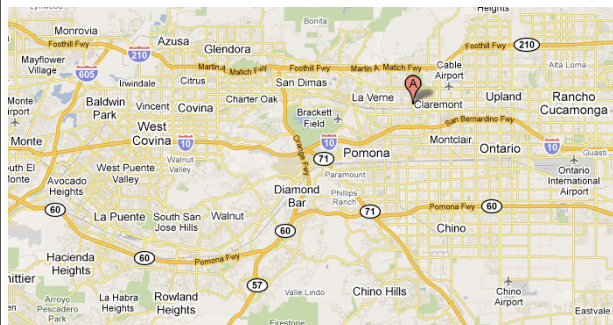
How do you think google maps does it?



35

## from: Claremont to: Rowland Heights

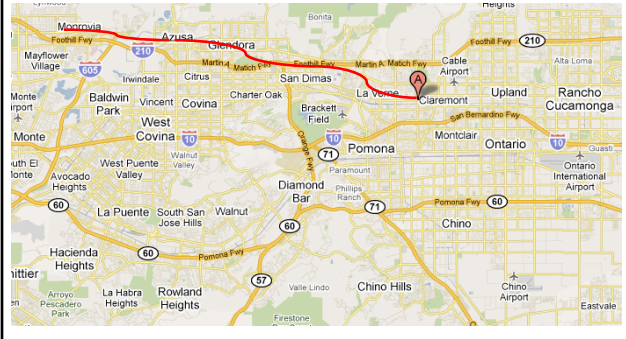
What would the search algorithms do?



36

from: Claremont to: Rowland Heights

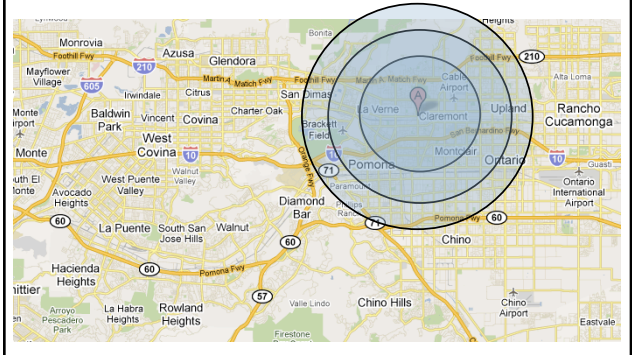
DFS



37

from: Claremont to: Rowland Heights

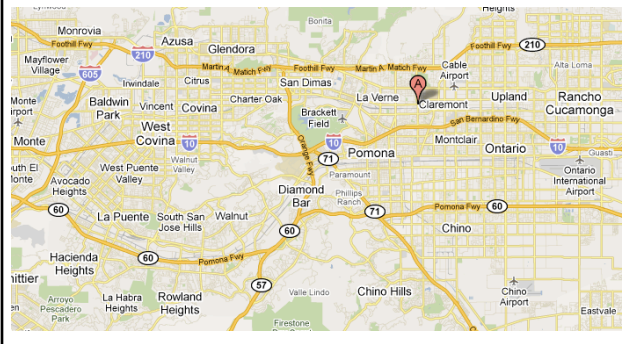
BFS



38

from: Claremont to: Rowland Heights

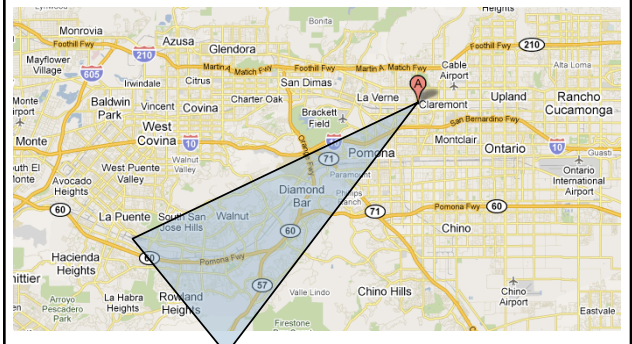
Ideas?



39

from: Claremont to: Rowland Heights

We'd like to bias search towards the actual solution



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  $h(n)$

41

## 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).

42

## 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

43