INFORMED SEARCH LAST THOUGHTS

David Kauchak
CS51A – Spring 2022

1

Admin

Assignment 9

Assignment 10

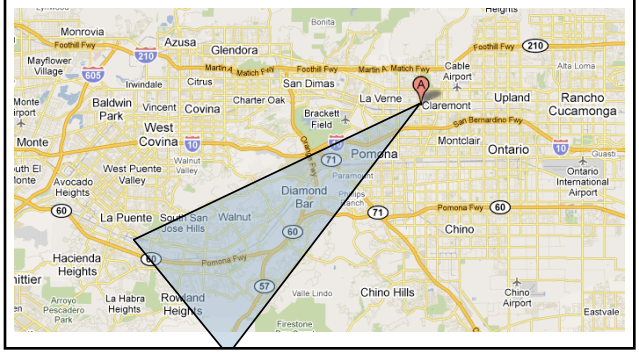
Midterm 2 next Monday

- Dictionaries (2/21) through today
- "cheat" sheet – two pages
- Sample problems posted

2

from: Claremont to: Rowland Heights

We'd like to bias search towards the actual solution



3

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)*

4

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

5

Two heuristics

2	8	3
1	6	4
	7	5

1	2	3
8	6	4
	7	5

6	2	3
8		4
7	1	5

Which state is better?

1	2	3
8		4
7	6	5

GOAL

6

Two heuristics

2	8	3
1	6	4
	7	5

1	2	3
8		4
7	6	5

Goal

What is the "distance" of the tiles that are out of place?

7

Two heuristics

2	8	3
1	6	4
	7	5

1	2	3
8		4
7	6	5

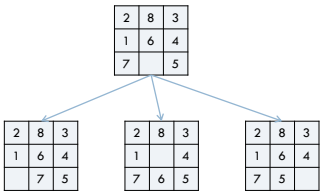
Goal

6

8

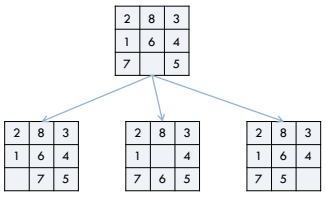
Next states?

9



Which would you do?

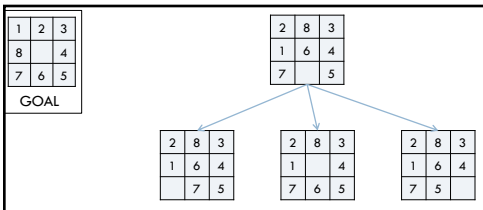
10



Which would DFS choose

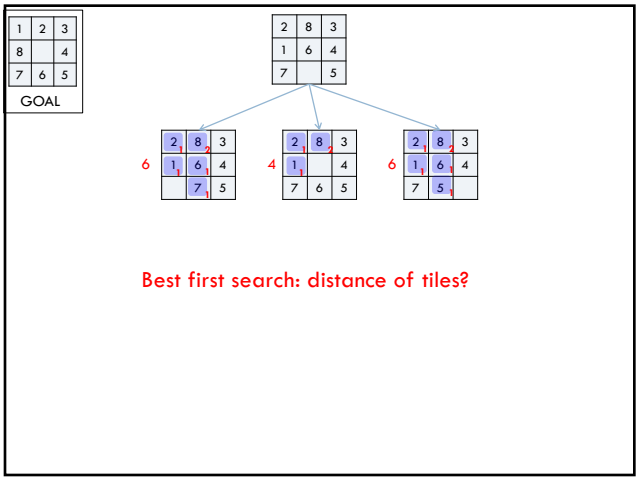
Completely depends on how next states are generated.
Not an "intelligent" decision!

11

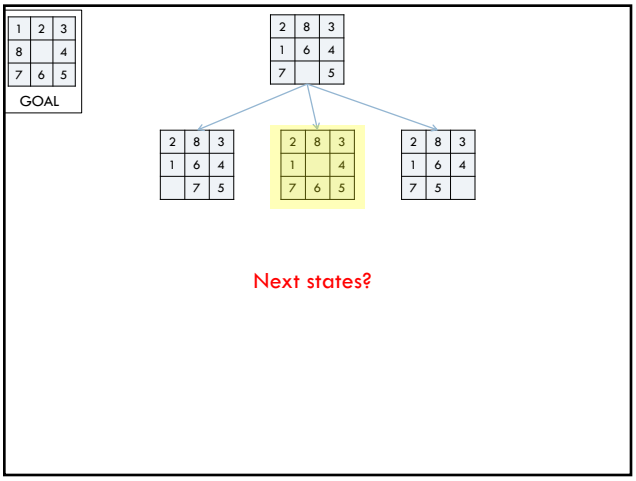


Best first search: out of place tiles?

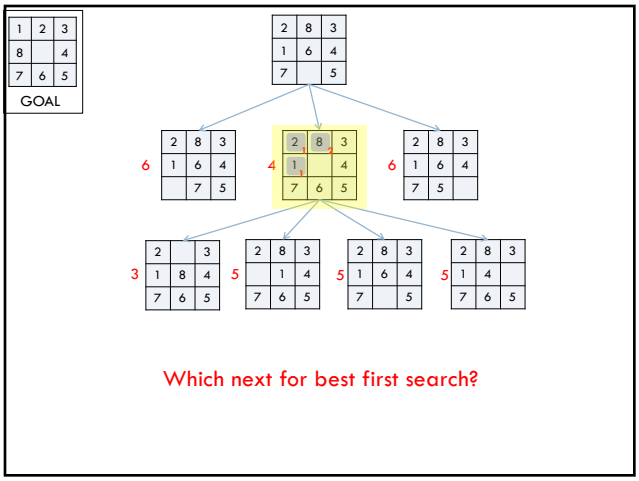
12


GOAL
Best first search: distance of tiles?

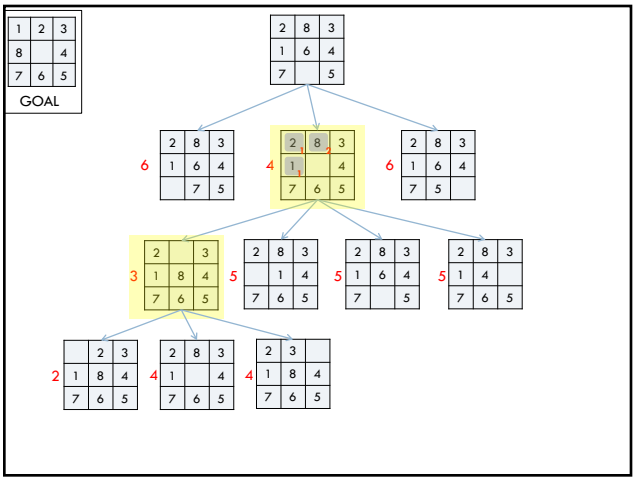
13


GOAL
Next states?

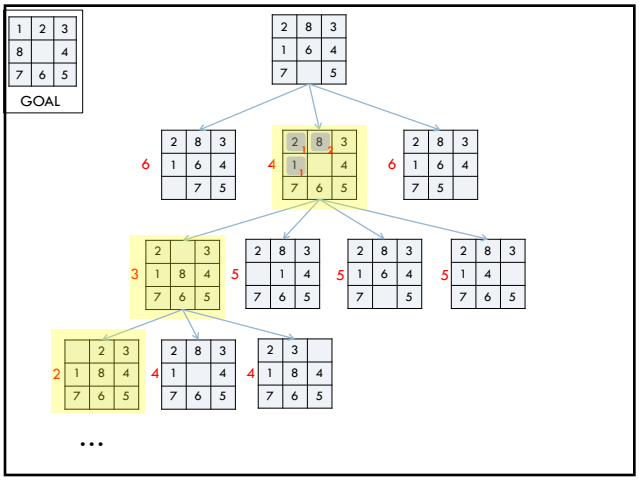
14


GOAL
Which next for best first search?

15


GOAL

16



17

Informed search algorithms

Best first search is called an "informed" search algorithm

Why wouldn't we always use an informed algorithm?

- Coming up with good heuristics can be hard for some problems
- There is computational overhead (both in calculating the heuristic and in keeping track of the next "best" state)

18

Informed search algorithms

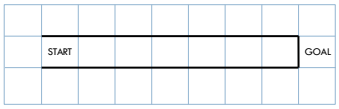
Any other problems/concerns about best first search?

19

Informed search algorithms

Any other problems/concerns about best first search?

- Only as good as the heuristic function



Best first search using distance as the crow flies as heuristic

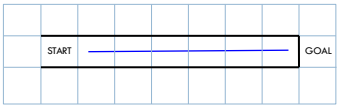
What would the search do?

20

Informed search algorithms

Any other problems/concerns about best first search?

- Only as good as the heuristic function



Best first search using distance as the crow flies as heuristic

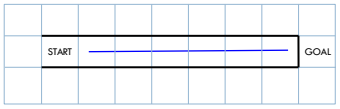
What is the problem?

21

Informed search algorithms

Any other problems/concerns about best first search?

- Only as good as the heuristic function



Best first search using distance as the crow flies as heuristic

Doesn't take into account how far it has come.
Best first search is a "greedy" algorithm

22

Informed search algorithms

Best first search is called an "informed" search algorithm

There are many other informed search algorithms:

- A* search (and variants)
- Theta*
- Beam search

23

Sudoku

	4	3				6	7	
5			4		2			8
8				6				1
2								5
	5						4	
		6				7		
			5		1			
				8				

Fill in the grid with the numbers 1-9

- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

24

Sudoku

7	2	8	9	3	6	5	1	4
9	4	3	1	5	8	6	7	2
5	6	1	4	7	2	9	3	8
8	3	4	7	6	5	2	9	1
2	1	7	8	4	9	3	6	5
6	5	9	2	1	3	8	4	7
1	8	6	3	2	4	7	5	9
3	7	2	5	9	1	4	8	6
4	9	5	6	8	7	1	2	3

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

25

Sudoku

	4	3				6	7	
5			4		2			8
8				6				1
2								5
	5						4	
		6				7		
			5		1			
				8				

How can we pose this as a search problem?

State

Start state

Goal state

State space/transitions

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

26

Sudoku

	4	3				6	7	
5			4		2			8
8				6				1
2								5
	5						4	
		6				7		
			5		1			
				8				

How can we pose this as a search problem?

State: 9 x 9 grid with 1-9 or empty

Start state:

Goal state:

State space/transitions

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

27

Sudoku

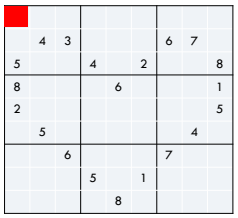
	4	3				6	7	
5			4		2			8
8				6				1
2								5
	5						4	
		6				7		
			5		1			
				8				

Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

28

Sudoku



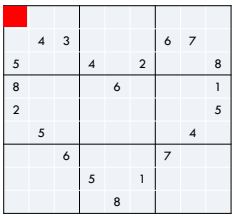
Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

How many next states?
What are they?

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

29

Sudoku



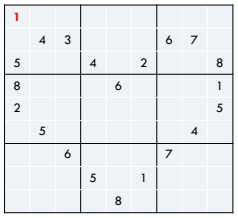
Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

1, 6, 7, 9

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

30

Sudoku



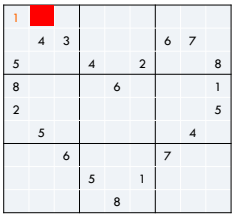
Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

1, 6, 7, 9

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

31

Sudoku



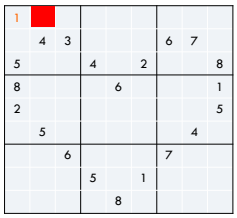
Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

How many next states?
What are they?

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

32

Sudoku



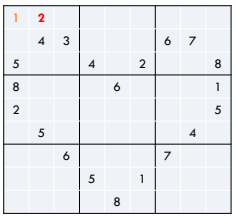
Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

2, 6, 7, 8, 9

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

33

Sudoku



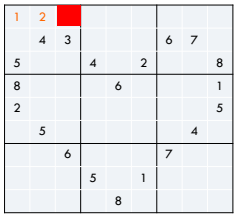
Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

2, 6, 7, 8, 9

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

34

Sudoku



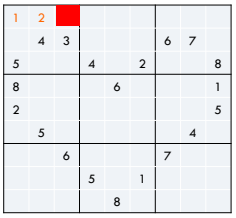
Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

What are the next states?

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

35

Sudoku



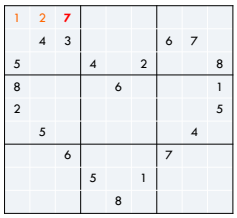
Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

7, 8, 9

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

36

Sudoku



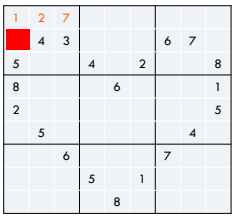
Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

7, 8, 9

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

37

Sudoku

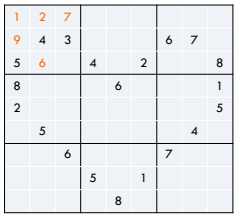


Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

38

Sudoku

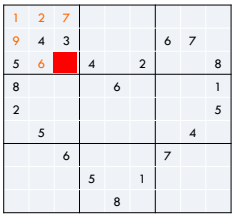


Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

39

Sudoku



Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

Now what?

Try another branch, i.e. go back to a place where we had a decision and try a different one

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

40

Sudoku

1	2	8						
	4	3				6	7	
5			4		2			8
8				6				1
2								5
	5						4	
		6				7		
			5		1			
				8				

Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

7, 8, 9

Fill in the grid with the numbers 1-9
- each row has 1-9 (without repetition)
- each column has 1-9 (without repetition)
- each quadrant has 1-9 (without repetition)

41

Best first Sudoku search

DFS and BFS will choose entries (and numbers within those entries) randomly

Is that how people do it?

How do you do it?

Heuristics for best first search?

Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

42

Best first Sudoku search

DFS and BFS will choose entries (and numbers within those entries) randomly

Pick the entry that is **MOST** constrained

People often try and find entries where only one option exists and only fill it in that way (very little search)

Generate next states:
- pick an open entry
- try all possible numbers that meet constraints

43

Representing the Sudoku board

	4	3				6	7	
5			4		2			8
8				6				1
2								5
	5						4	
		6				7		
			5		1			
				8				

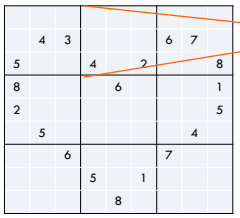
Board is a matrix (list of lists)

SudokuEntry: Class to keep track of an entry on the sudoku board

- Keep track of whether or not a value has been placed in that entry (fixed)
- If one hasn't, keep track of what numbers are still valid

44

Representing the Sudoku board



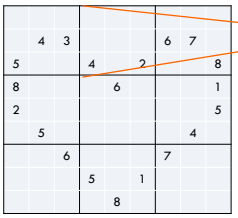
[1, 6, 7, 9], [1, 2, 6, 7, 8, 9], [1, 2, 7, 8, 9],
[1, 9], 4, 3,
5, [1, 6, 7, 9], [1, 7, 9]

- Board will consist of 81 SudokuEntry s
- Each quadrant will have 9 entries

- Board is a matrix (list of lists)
- Each entry is a SudokuEntry:
  - "fixed" if it has a number place
  - if not, then keeps a list of values still available

45

Representing the Sudoku board



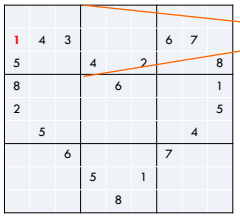
[1, 6, 7, 9], [1, 2, 6, 7, 8, 9], [1, 2, 7, 8, 9],
[1, 9], 4, 3,
5, [1, 6, 7, 9], [1, 7, 9]

Which is the most constrained (of the ones above)?

- Board is a matrix (list of lists)
- Each entry is a SudokuEntry:
  - "fixed" if it has a number place
  - if not, then keeps a list of values still available

46

Representing the Sudoku board



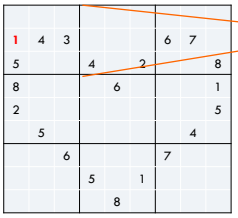
[1, 6, 7, 9], [1, 2, 6, 7, 8, 9], [1, 2, 7, 8, 9],
[1, 9], 4, 3,
5, [1, 6, 7, 9], [1, 7, 9]

What would the state look like if we add pick 1?

- Board is a matrix (list of lists)
- Each entry is a SudokuEntry:
  - "fixed" if it has a number place
  - if not, then keeps a list of values still available

47

Representing the Sudoku board



[6, 7, 9], [2, 6, 7, 8, 9], [2, 7, 8, 9],
[9], 4, 3,
5, [6, 7, 9], [7, 9]

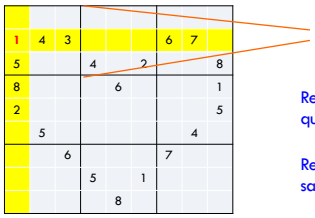
Remove 1 from all entries in the quadrant

What other parts of the board need to be updated?

- Board is a matrix (list of lists)
- Each entry is a SudokuEntry:
  - "fixed" if it has a number place
  - if not, then keeps a list of values still available

48

Representing the Sudoku board



[6, 7, 9], [2, 6, 7, 8, 9], [2, 7, 8, 9],
[9], 4, 3,
5, [6, 7, 9], [7, 9]

Remove 1 from all entries in the quadrant

Remove 1 from all entries in the same column

Remove 1 from all entries in the same row

- Board is a matrix (list of lists)
- Each entry is a SudokuEntry:
  - "fixed" if it has a number place
  - if not, then keeps a list of values still available

49