Path-Finding Algorithms

EN 500.111 Week 2

Outline

➜   What is an algorithm?
➜   Navigating through a grid
➜   Generalizing to other graphs

This is the first of two lectures on pathfinding

➜   *This week*: Focus on some core theoretical algorithms
➜   *Next week*: Focus on practicality and finding routes for navigation

What is an algorithm?

➔ **Definition**: a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
➔ Can refer to many different types of processes, both within and outside of computing
➔ Generally 3 main components:
- ◆ Inputs
- ◆ Series of steps
- ◆ Outputs

Example: Recipe for chocolate chip cookies

Inputs	Series of Steps	Outputs
1 cup chocolate chips 1.5 cups flour .75 cups sugar etc.		2 dozen cookies

https://simple-veganista.com/vegan-chocolate-chip-cookies/

What are some other examples of algorithms you might use?

Changing the Input

➜ You can change the input amount - e.g., by doubling everything - and get twice as much output **without changing the steps**

➜ But certain parts of the recipe might take longer - we'll revisit this when we talk about runtime analysis

Some recipe websites automate this part for you!

INGREDIENTS
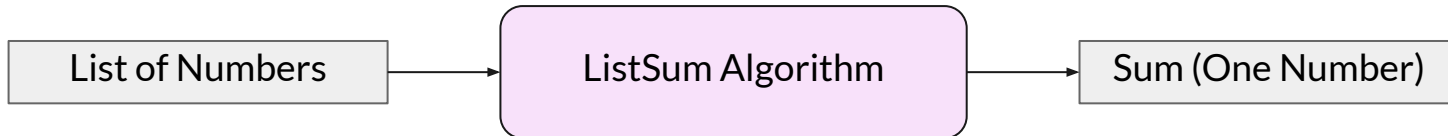
SCALE 1X 2X 3X

Another Algorithm: Adding Numbers in a List

ListSum Algorithm:

➔ **Input**: A list of numbers
➔ **Steps**:
   1. Start with a sum of 0
   2. Read the first number from the list, add it to the sum, and cross it out
   3. If the list is now empty, output the sum. Otherwise, go to step 2.
➔ **Output**: A number equal to the sum of numbers in the list

| List of Numbers | → | ListSum Algorithm | → | Sum (One Number) |

Example of ListSum Algorithm

ListSum Algorithm:

Steps:

1. Start with a sum of 0
2. Read the first number from the list, add it to the sum, and cross it out
3. If the list is now empty, output the sum. Otherwise, go to step 2.

Input: [1, 2, 3, 4]

Step 1: List = [1, 2, 3, 4], Sum = 0
Step 2: List = [~~1~~, 2, 3, 4], Sum = 1
Step 3: List still not empty, so repeat step 2
Step 2: List = [~~1, 2~~, 3, 4], Sum = 3
Step 3: List still not empty, so repeat step 2
Step 2: List = [~~1, 2, 3~~, 4], Sum = 6
Step 3: List still not empty, so repeat step 2
**Step 2**: List = [~~1, 2, 3, 4~~], Sum = 10
**Step 3**: List is empty, so output 10

Output: 10

Algorithms on Lists

➔ In this example, we made an algorithm that takes a data structure (a list of numbers) and produces the correct output regardless of how many numbers we have or what they are

➔ There are a lot of other algorithms we might want to write with a list as input:
  ◆ The most frequent number in the list
  ◆ The median of a list
  ◆ The pair of numbers with the smallest difference between them
  ◆ Others?

```
List of Numbers  →  Algorithm  →  Some Output
```
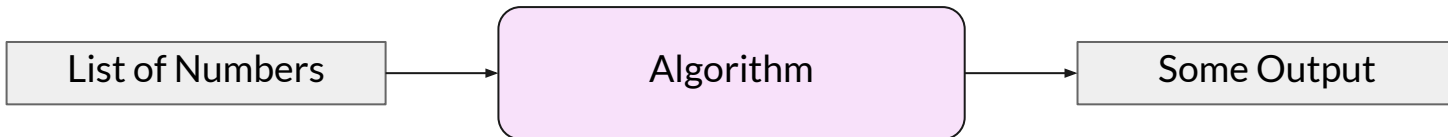
Finding a Path through a Grid
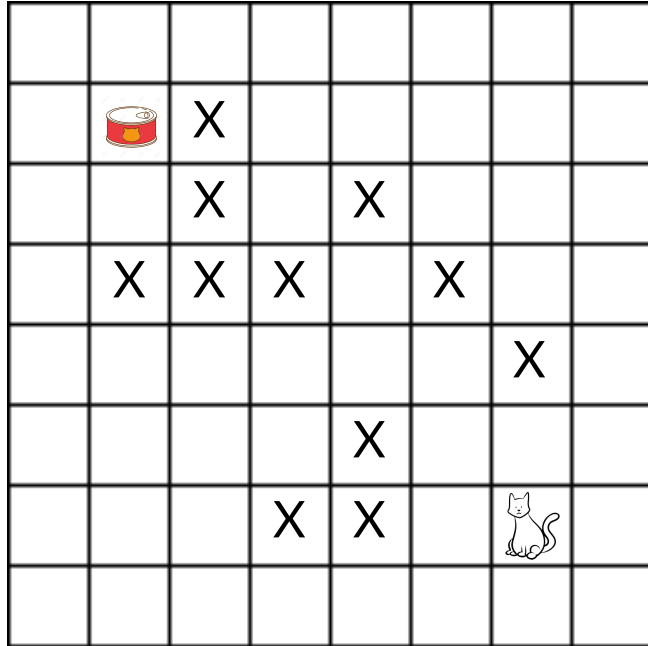
➜ The cat in this maze can move up, down, left, or right at one square per second.

➜ The squares marked with an X are blocked off and the cat cannot travel through them.

➜ What's the fastest time in which they can reach the cat food?
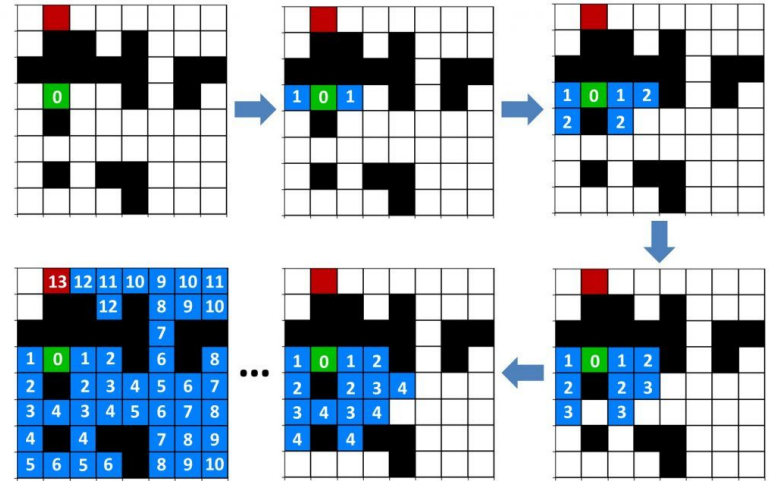
Harder Version - Specialized Movement

https://www.mark-weeks.com/aboutcom/ble132kn.htm
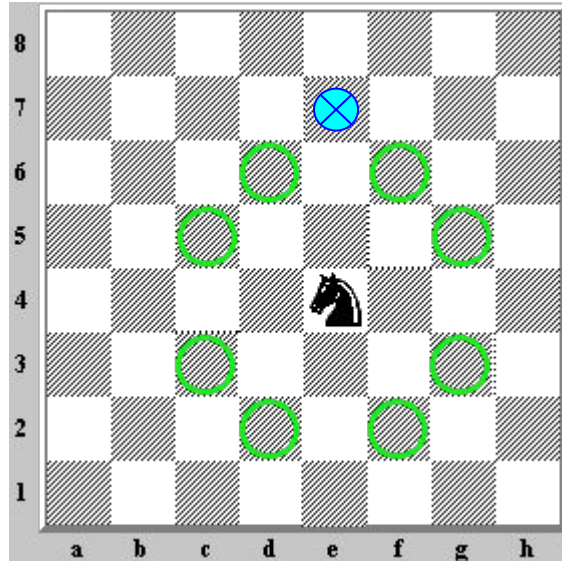
Knights in chess can move in an "L-shape" - 2 squares in one direction and 1 square in another perpendicular direction.

How many moves would it take the knight pictured here to get to the blue X?

Pathfinding Algorithm: Breadth-First Search (BFS)

➔ **Inputs**: A 2-D grid, markings of which squares are blocked off, a starting point, and an ending point
➔ **Steps**:
  1. Mark the starting point with a 0.
  2. Mark any non-blocked squares next to a 0 square with a 1.
  3. Mark any non-blocked squares next to a 1 square with a 2.
  4. Continue as in steps 2 and 3 until the end square has been marked.
  5. Build a path from the end back to the start, repeatedly going to a square with a number which is one smaller than the previous one.
➔ **Output**: A path connecting the start and the end

Cat BFS Example

Cat BFS Example

Cat BFS Example

Cat BFS Example

	🥫	X					
		X		X			
	X	X	X		X		4
				4	3	X	3
				X	2	1	2
			X	X	1	🐈	1
			4	3	2	1	2

Cat BFS Example

	🥫	X					
		X		X			5
	X	X	X	5	X	5	4
			5	4	3	X	3
				X	2	1	2
			X	X	1	🐱	1
		5	4	3	2	1	2

Cat BFS Example

	🥫	X					6
		X		X		6	5
	X	X	X	5	X	5	4
		6	5	4	3	X	3
			6	X	2	1	2
		6	X	X	1	🐱	1
	6	5	4	3	2	1	2

Cat BFS Example

							7
	🥫	X				7	6
		X		X	7	6	5
	X	X	X	5	X	5	4
	7	6	5	4	3	X	3
		7	6	X	2	1	2
	7	6	X	X	1	🐱	1
7	6	5	4	3	2	1	2

Cat BFS Example

						8	7
	🥫	X			8	7	6
		X		X	7	6	5
	X	X	X	5	X	5	4
8	7	6	5	4	3	X	3
	8	7	6	X	2	1	2
8	7	6	X	X	1	🐱	1
7	6	5	4	3	2	1	2

Cat BFS Example

					9	8	7
	🥫	X		9	8	7	6
		X		X	7	6	5
9	X	X	X	5	X	5	4
8	7	6	5	4	3	X	3
9	8	7	6	X	2	1	2
8	7	6	X	X	1	🐱	1
7	6	5	4	3	2	1	2

Cat BFS Example

				10	9	8	7
	🥫	X	10	9	8	7	6
10		X		X	7	6	5
9	X	X	X	5	X	5	4
8	7	6	5	4	3	X	3
9	8	7	6	X	2	1	2
8	7	6	X	X	1	🐱	1
7	6	5	4	3	2	1	2

Cat BFS Example

			11	10	9	8	7
11	🥫	X	10	9	8	7	6
10	11	X	11	X	7	6	5
9	X	X	X	5	X	5	4
8	7	6	5	4	3	X	3
9	8	7	6	X	2	1	2
8	7	6	X	X	1	🐱	1
7	6	5	4	3	2	1	2

Cat BFS Example

12		12	11	10	9	8	7
11	12	X	10	9	8	7	6
10	11	X	11	X	7	6	5
9	X	X	X	5	X	5	4
8	7	6	5	4	3	X	3
9	8	7	6	X	2	1	2
8	7	6	X	X	1		1
7	6	5	4	3	2	1	2

We finally marked the destination, so now we can trace back the path!

Cat BFS Example

		12	11	10	9	8	7
	12	X	10	9	8	7	6
10	11	X	11	X	7	6	5
9	X	X	X	5	X	5	4
8	7	6	5	4	3	X	3
9	8	7	6	X	2	1	2
8	7	6	X	X	1	🐈	1
7	6	5	4	3	2	1	2

So the cat can reach the food in twelve seconds with this path: ULULLLLLUURU
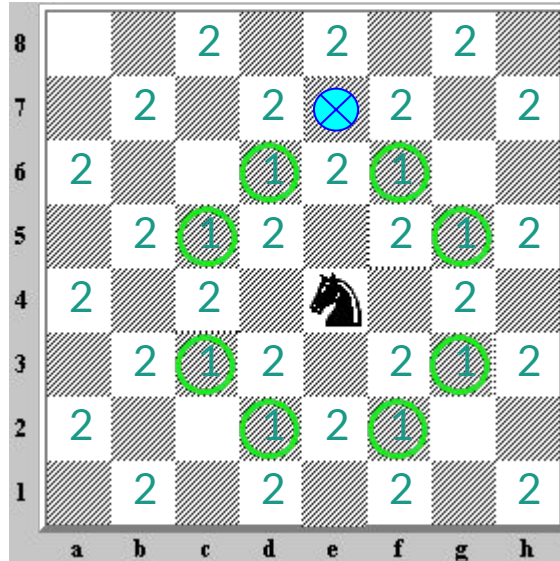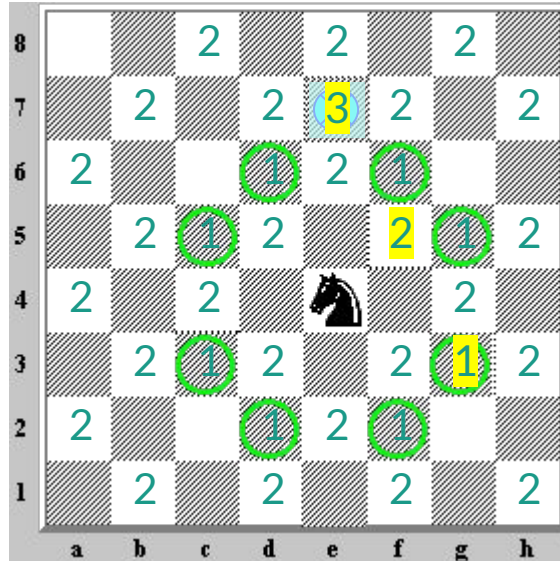
Knight Version of BFS
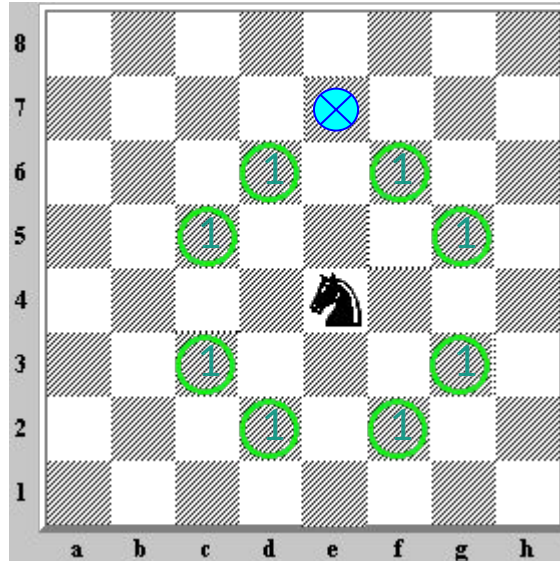
Same idea but "adjacent" now means anywhere the knight can move from the current square
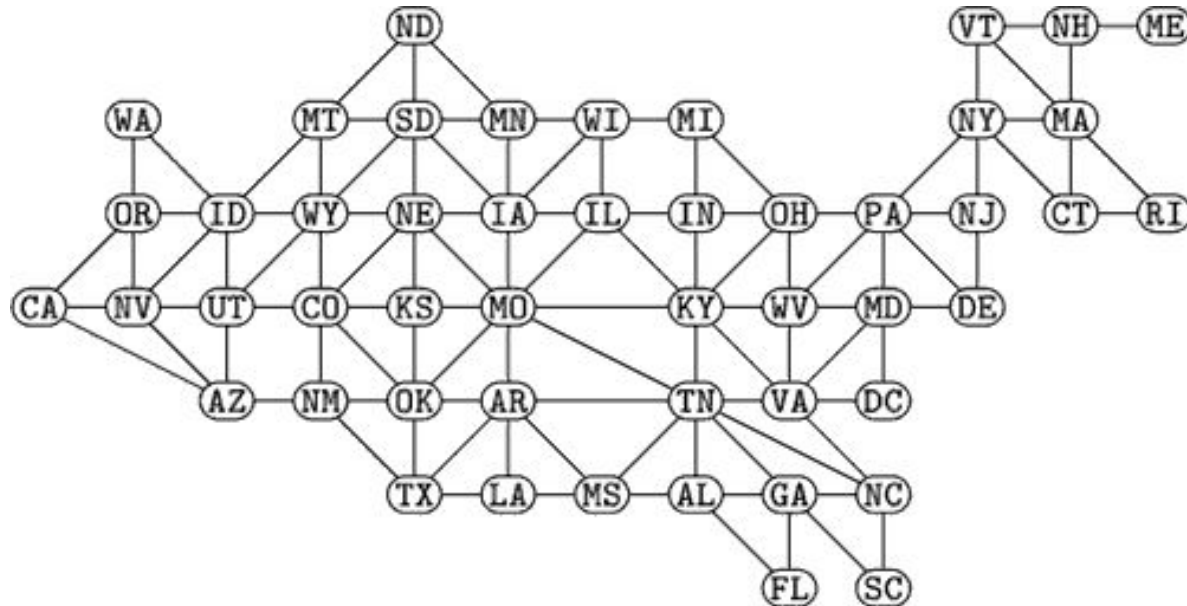
Knight Version of BFS

Since there are up to 8 options for each move, the possible paths branch out very quickly and much of the board can be reached in at most 2 moves

Knight Version of BFS

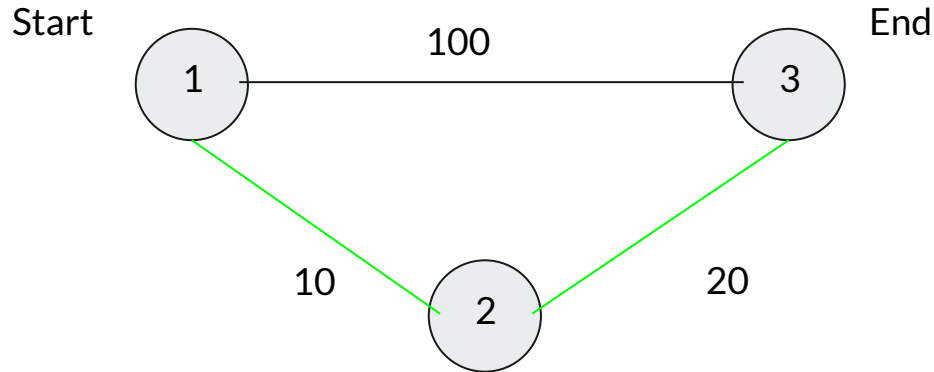There are a lot of ways to get to the destination in 3 moves, but one path is highlighted.

Application to non-grid graphs

How many of the 48 continental US states (plus DC) can you reach from MD by crossing at most 3 state borders?
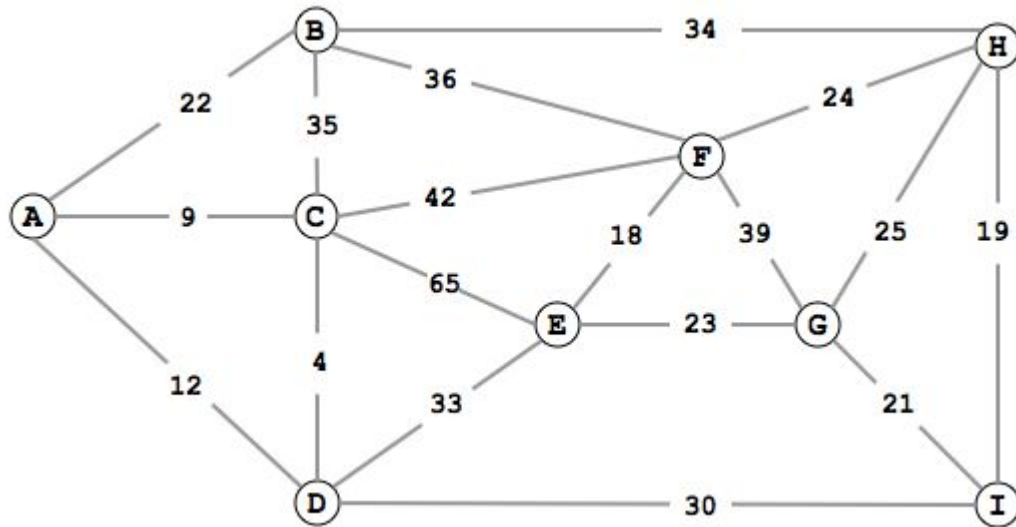
Discuss in groups!

Shortest Path in Weighted Graphs

Start

End

100

1

3

10

2

20

We want the path with the shortest total edge weight, which is not necessarily the path with the fewest edges

Shortest Path Example

What's the shortest path from A to G in this graph?

Extending BFS to Weighted Graphs

➔ BFS has the important property that it groups nodes by how far away they are from the start and processes them in increasing order
  ◆ First vertex processed is the start, which is 0 edges away
  ◆ Then is everything one edge away (marked with a 1)
  ◆ Then everything two edges away (marked with a 2)
  ◆ And so on
➔ **Goal**: do something similar for weighted graphs where we still process nodes in increasing order of distance

		12	11	10	9	8	7
	12	X	10	9	8	7	6
10	11	X	11	X	7	6	5
9	X	X	X	5	X	5	4
8	7	6	5	4	3	X	3
9	8	7	6	X	2	1	2
8	7	6	X	X	1	🐱	1
7	6	5	4	3	2	1	2

Any questions?