Opening Discussion

- IcP solutions.
- Minute essay comments:
  - Why do I teach the easy stuff after the hard stuff?
  - Could you index a map by a case class?
  - Buffer: the most imperative structure.
Previously we used recursion to create iteration. This is done with a recursive method that calls itself once and can often be done better with loops.

The real power of recursion comes in when the method calls itself two or more times.

The call stack provides memory so recursion can do one thing, then come back and do another.
Fibonacci Numbers

- The simplest example of a recursive function that calls itself more than once is the Fibonacci numbers.
  - 1, 1, 2, 3, 5, 8, 13, 21, ...
- Each number is the sum of the two before it.
  - $f(n) = \text{if}(n>2) \ f(n-1) + f(n-2) \ \text{else} \ 1$
- Simple, but not great.
Towers of Hanoi

- A classic example of recursion is solving the Towers of Hanoi.
- This game is generally made with disks and three pegs.
- You need to move the disks from one peg to another.
  - Can only move one disk at a time.
  - Can't place a disk on one smaller than it.
- Solution to N disks: move N-1 disks, move 1 disk, move N-1 disks.
My favorite example is mazes.

Consider a maze as a 2-D grid with each square either filled or not.

Now the challenge is to find the length of the shortest path through the maze.

How do you do that?
Minute Essay

- What questions do you have about stuff?
- What are your Thanksgiving plans?
- Interclass Problem:
  - Write a recursive function that calls itself twice. Made it so it recurses down n levels where n is the argument you pass in the first time you call it. Have the function keep track of how many total times it is called.