Algorithms

Iteration and Recursion

Maurice

Phase 1: Day 3

What is an Algorithm?

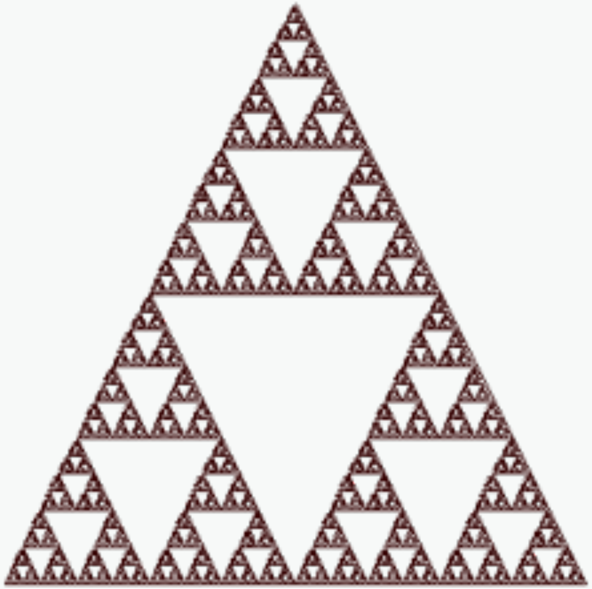
- Steps to follow to solve a problem

Different Approaches

- Iterative solutions
- Recursive solutions

What are Recursive Solutions?

Solving a problem by combining the solutions of *smaller instances* of the *same* problem



Characteristics of Recursion

- Base case: the smallest instance whose solution is known
- The method calls itself

I DON'T ALWAYS USE RECURSION
BUT WHEN I DO, I DON'T ALWAYS USE RECURSION

Place in line Demo

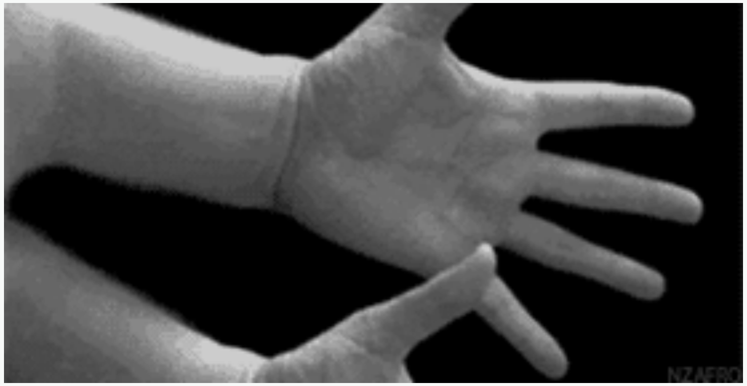
- *What place in line am I?*
  - Iteratively
  - Recursively

Recursive solution

```
Hey what place in line are you?
  if  Am I the first in line?
    answer ☞  1
  else
    ask neighbor in front of me, Hey what place in line are you?
    answer ☞  the neighbor’s answer + 1
  end
```

Reverse a list demo

- Plates demo
- Review execution of *iterative* version
- Live code *recursive* version
- Demystify the *magic*



The Recursive mindset

- Faithful
- Timid
- Lazy
- Elegant

Beware *madness!*

Infinite recursion!

Palindromes

- Words that are spelled the same forward and backward

civic
deified
redder

Detect Palindromes Recursively

- In your groups, write pseudocode for a recursive method `is_palindrome?`

```
def palindrome?(word)
end

palindrome?(“ruby”)
# => false

palindrome?(“civic”)
# => true
```