Recursion
Ch 9.1

In order to understand recursion, you must understand recursion.
Highlights

- recursion

```java
public static void main(String[] args)
{
    main(null);
}
```
There are two important parts of recursion:
- A stopping case that ends the recursion
- A reduction case that reduces the problem

What are the base and stopping cases for the Fibonacci numbers?

\[ F_n = F_{n-1} + F_{n-2}, \]
\[ 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \ldots \]

(sum of the previous two numbers)
(see last time: FibonacciRecursion.java)
Recursion

What if I wanted to just count down to zero? `countdown(5)` would show:

5
4
3
2
1
0!

(see: Countdown.java)
Recursion: Basic example

Remember, code starts in main and runs from top to bottom in sequence (normally)

When you call a function you go execute all the function's code is run before going back to the original code

Code order is important in recursion!

(See: StringRecursion.java)
Recursion: Root finding

Find a root of:
(see: RootFind.java)

Method:
1. Find one positive $y$ and 1 neg. $y$
2. Find midpoint (of x values)
3. update $y$-pos/neg
Recursion: Dictionary search

Open the dictionary to the middle
- If the word is not on that page, reopen in the middle of the unsearched side

(See: DictionarySearch.java)
Recursive delete

The File class can (regarding delete)...
- Delete only a single file at a time
- A directory must be empty in order to be deleted

Write a program that would delete all files and all directories

(See: RecursiveDelete.java)
Recursion

How would you sum the numbers 1 to n using recursion (not a loop)?

For example sumToN(5) = 15, as 1+2+3+4+5 = 15

What is the stopping case?
How do you reduce the problem?

(see: SumToN.java)
Recursion

What if we defined tangent recursively as:

$$\tan(x) = \frac{x}{1 - \frac{x^2}{3 - \frac{x^2}{5 - \frac{x^2}{7-\ldots}}}}$$

Assume we take an input for how many times to do this recursion

What is the pattern? What is the stopping case?

How do we move towards the stopping case

(see: Tangent.java)
Recursion

The tower of Hanoi is played by:
1. Moving a single ring to another stack
2. Smaller rings cannot have larger rings on top of them

(see: TowerHanoi.java)
Recursion

Do not try to solve chess in this manner!

You will segfault
(you will also not finish computing before the sun burns the earth to a crisp)