

LEC 09

CSE 123

Recursive Programming

BEFORE WE START

Talk to your neighbors:

Coke, Pepsi, Dr. Pepper, or None of those?

Instructor: James Wilcox

Questions during Class?

Raise hand or send here

sli.do **#cse123**

Announcements

- Resubmission Period 2 due tonight (10/24) at 11:59pm
 - Last opportunity for C0
- Quiz 1 Tuesday (10/28) in your registered section
- Programming Assignment 1 is due Wednesday (10/29) at 11:59pm

Recursive Methods [Review]

- 2 components of every recursive method:
- Recursive case
 - Decompose problem into subproblem
 - Make the actual recursive call
 - Combine results meaningfully
- Base case
 - Simplest version of the problem
 - No subproblems to break into
 - Return known answer



If decomposing moves you closer to the base, no infinite recursion!

Why Recursion?

- Generally, anything you can write iteratively you can write recursively
 - So why write anything recursively?

Recursion is particularly useful when dealing with something that's recursively defined

- Math examples:
 - Factorial: $n! = n * (n - 1)!$
 - Exponent: $x^n = x * x^{n-1}$
 - Fibonacci: $fib(n) = fib(n - 1) + fib(n - 2)$
- Non-math examples?
 - ListNodes (int data, ListNode next)
 - Other ideas?

Public / Private Pairs

- Used when we need additional information between recursive calls
- Private helper method hides additional info
 - Clients shouldn't have to worry about it
- All public method does is kick-start the private one
 - What's the correct starting value(s) for additional param(s)?

Question to ask: "Do I need to keep track of any additional information?"