

Slide 1

Administrivia

- Reminder: Quiz 5 Thursday. Likely topic is recursion.
- Reminder: Homework 5 code due Thursday. Due dates for remaining homeworks posted.
- No office hours Thursday, but I will be around Friday afternoon.

Slide 2

Recursion — Overview

- Basic approach:
 - Identify “base case” — something you can solve directly.
 - Figure out how to decompose non-base cases into “smaller” problems, and apply algorithm to smaller problems.
- How to think about “does it work?”
 - Does it work for base case(s)?
 - Assuming recursive calls work, does it work for other cases?
 - Does every recursive call get you at least one step closer to a base case?
- Implementation — conceptually (and usually in fact) involves a stack of calls-in-progress.
- Can be slower than iteration (though sometimes not), but can also be much easier to understand.

Slide 3

Recursion — Simple Examples

- Factorial function.
- Function to compute Fibonacci numbers (very slow!).

Slide 4

Recursion — More Examples

- Linked list implementation.
- Quicksort — pick “pivot” element, split array into elements less than pivot and elements greater than pivot, and sort recursively. Why does this work?
- Mergesort — split array (or list) into two pieces of equal size, sort recursively, merge. Why does this work?
- (Other example(s) as time permits.)

Minute Essay

- Why is the recursive implementation of the Fibonacci function so slow?

Slide 5

Minute Essay Answer

- It does quite a lot more work than an iterative version — a lot of duplication of effort.

Slide 6