

Slide 1

Administrivia

- Reminder: Homework 3 code due today.
- Reminder: Quiz 3 Thursday. Likely topic is stacks and queues.
- Reminder: Homework 4 design due Thursday.

Slide 2

Recap — ADTs, Array-Based Stacks and Queues

- Abstract data type — define possible “values”, list of operations. Could capture as Java `interface`.
- Stack ADT — LIFO queue.
- Queue ADT — FIFO queue.
- Array-based implementations:
 - Stacks easy, queues somewhat trickier (“circular queue”).
 - General approach — decide what variables we need, what they should “mean”.
 - Error checking — Java-esque way is to use exceptions.
 - Include `main` method for simple error checking.

Slide 3

Lists

- List ADT:
 - “Values” are lists of elements.
 - Many operations possible — add element, remove element, search for element, etc., etc.
(Also “walk through elements” with “iterator” — next time.)
- Implementation:
 - Using an array.
 - Using a “linked list”.

How do these compare with regard to efficiency of various operations?
efficiency of memory use?

Slide 4

Linked Lists

- Think about implementing some basic list operations (add, remove, find) using a linked list. First, draw pictures . . .
- Then think about what you need to turn the pictures into code. Probably you’ll need:
 - Variables (e.g., something to point to the first “node” (little box).
 - Classes-within-the-class (for nodes / little boxes, iterators).
 - Methods for interface.
- (Write code.)

Minute Essay

- None — sign in.

Slide 5