

1  CS3134 #11

10/7/03

Janak J Parekh

2  Administrivia

- Should get a new TA by Wednesday, hopefully
- Homework 2 problem(s) 3(b) and 3(c)
- Makeup OH on Wednesday because Andy is gone?
- Homework 1 without any name...
  - You'll get HW1s back on Thursday

3  Agenda

- Finish Linked Lists
- Start Recursion
- But first...
  - Meaning of the “.” operator, reviewed
  - What does `Last.prev.next = null` mean?

4  Other linked-list considerations

- Sorted List: how to do?
  - Cases when inserting at beginning, middle, or end
- Sorting an unsorted List
  - Insertion sort is faster than the other two sorts, since “sliding” is very easy to do
- Doubly-linked lists
  - Advantage: easy to go backwards
  - Disadvantage: more memory and bookkeeping

5  Iterators

- With lists, frequently need to walk through a list
  - Increase minimum wages of all employees, etc.
- But there's no array index! How to step through?
- One way is to keep references to current cell, but requires “outsider” to know the internals of how the list works
- Would be nice to have something like `StringTokenizer`'s “`nextToken()`” call

6  Iterators (II)

- Structure: list, current, and previous references
- Methods – book suggests:
  - `reset()` – go back to beginning
  - `nextLink()`
  - `getCurrent()`
  - `atEnd()` – *last* element, not after it
  - `insertAfter()`
  - `insertBefore()`
  - `deleteCurrent()`
- Java has its own, simpler, `Iterator`, with `next()` and `hasNext()`, and that's it
  - Supports more than linked lists

## 7 Iteration vs. Recursion

- So, what is iteration, anyway?
  - Dictionary.com: “The process of repeating a set of instructions a specified number of times or until a specific result is achieved.”
- Any other way of repeating over and over?
- Well, let’s think about it...

## 8 How to calculate...

- What’s the sequence 1, 3, 6, 10, 15, 21, 28, 36...
  - *Triangle* numbers
  - How to do as loop?
  - How to do *as addition on previous result?*
    - Recursion!

## 9 Next time...

- Recursion