Design problems, plus Lists and Iterators

2/15/16
Design problems

• Figure out what operations are needed to support a story

• Identify data structures and implementations to efficiently provide those operations
What data structure would you use?

A. Set
B. Stack
C. Queue
D. Deque
E. Priority Queue
Announcements

• Reading:
  – Wednesday: Java Interlude 7 and “Searching a sorted array” in Chapter 18 (pp. 532-539)
  – Friday: Java Interlude 8 and Chapter 19 (Dictionaries)

• Homework due Tuesday night
  – Sorting nearly-sorted arrays and GUI layout design
List ADT

- void add(T)  //add to end
- void add(int, T)  //add at given position
- T remove(int)  //remove from given position
- void clear()
- T replace(int, T)
- T get(int)
- int size()
- ...
Adding a tail pointer

“Tail pointer” (tail reference) refers to last Node in the list
Adding a tail pointer

“Tail pointer” (tail reference) refers to last Node in the list
Which of the following correctly adds to the end?
A. tail = new Node(value, null);
B. tail = new Node(value, tail);
C. tail.next = new Node(value, null);
D. tail.next = new Node(value, tail.next);
E. Not exactly one of the above
“Tail pointer” (tail reference) refers to last Node in the list.
Which of the following correctly adds to the end?
A.  \( \text{tail} = \text{new Node(value, null)}; \)
B.  \( \text{tail} = \text{new Node(value, tail)}; \)
C.  \( \text{tail.next} = \text{new Node(value, null)}; \)
D.  \( \text{tail.next} = \text{new Node(value, tail.next)}; \)
E.  Not exactly one of the above (C, but must update tail)