Lecture 12: Stacks

CS 62
Spring 2013
Kim Bruce & Kevin Coogan

You never know when CS will be relevant!

Obama at Google

Weekly Lab

- Lab: JUnit
  - Unit testing with Java. Learn how to generate complete set of test for each method in program.

Weekly Assignment

- Assignment: Compression
  - Need to define new class CurDoublyLinkedList
    - Keeps track of "current" elt.
    - Can be subclass of DoublyLinkedList from Structure5 library.
  - Get up to two points extra credit if turn in design by Thursday midnight.

Stack

- Interface Stack<E> {
  - void push(E value)
  - E pop()
  - E peek()
- Example: Trays in cafeteria

- Last In - First Out (LIFO)

Stack Applications

- Run-time stack:
  - See sum program
- Backtracking
  - Solving Maze
- Evaluating expression in postfix form:
  - (52 - ((5 + 7) * 4) ⇒ 52 5 7 + 4 * - ⇒ 4
- Tools to parse programs

Stack Implementations

- ArrayList:
  - Which end should be head?
  - How complex for push, pop, peek?
- SinglyLinkedList: *Why not doubly-linked?*
  - Which end should be head?
  - How complex for push, pop, peek?
- Space differences?
  - What if there are several stacks?
- java.util.Stack based on Vector - don't use!

Queue

- FIFO: Waiting in line
- Operations:
  - enqueue (at end)
  - dequeue (from beginning)
- Examples:
  - Simulations
  - Event queue
  - Keeping track when searching

Queue Implementations

- SinglyLinkedList:
  - Which end should be front, rear?
  - How complex for enqueue, dequeue?
- ArrayList:
  - Which end should be front, rear?
  - How complex for enqueue, dequeue?
- Space differences?