Stacks and Queues

11/30/2007

Opening Discussion

- Let's look at solutions to the interclass problem.
- Linked list code.
- Do you have any questions about the assignment?

Stack and Queue ADTs

- Today we are going to talk about two other ADTs that are far simpler than lists.
- The stack and queue ADTs each require two methods (and generally have four)
- Stack
  - push – add something to the stack
  - pop – remove from the stack
- Queue
  - enqueue – add something to the queue
  - dequeue – remove something from the queue
- Both
  - peak – check next this to remove
  - isEmpty – tell if it is empty

LIFO vs. FIFO

- The difference between the stack and the queue is what element is removed when an element is removed.
- The stack is “Last In, First Out”.
- The queue is “First In, First Out”.
- Note you have no control over where things go or what you pull out. The order is specified by the ADT.

Array Based Stacks

- This is an easy data structure. Just keep an array and an integer index for the “top”.
- When the array fills up make a bigger one.

Array Based Queues

- This is a bit harder. You have to keep track of both a “front” and “back” of the queue. The trick is making it circular so you don't take up too much memory.
- Modulo is a very helpful operation for making the indexes wrap around.

Linked List Based Stacks

- Just as easy as the array based stack.
- Use a singly linked list and add and remove from the head.

Linked List Based Queues

- Perhaps easier than the array based queue.
- Use a singly linked list and make the head be the “front” while the tail is the “end”.
- It has to go that way because you can't efficiently remove from the tail in a singly linked list.

Code

- Let's code these up as time allows.

Minute Essay

- Do the linked data structures make sense to you?
- Interclass Problem – Look up what a reverse Polish calculator is and write one using a stack that holds doubles.