List Based Stacks and Queues

3/6/2008

Opening Discussion

- Let's look at solutions to the interclass problem.
- Iterators and how they work.

Stacks/Queues with Linked Lists

- The last time we talked about stacks and queues we implemented the ADTs using arrays. Part of the idea of an ADT though is that the implementation can vary as long as it has the right behavior.
- Since you now know how to write linked lists, we want to look at how we would write stacks and queues using linked lists.

Code

- I want to write a ListStack and a ListQueue. These should implement MyStack and MyQueue, but instead of using arrays (like ArrayStack and ArrayQueue), they should have a linked list inside of them.
- Remember that it is essential that the implementation be O(1) for all operations.

Priority Queue ADT

- A priority queue has the same methods as a normal queue, only the contents are ordered not only be arrival time, but also by a priority. So dequeue gets the highest priority object and if several have that priority, it gets the one that has been there the longest.
- One way to implement a priority queue is with a sorted linked list. To make this flexible, you could have it take a comparator that tells you the ordering. That would be provided when the priority queue is constructed.
- What order are the various operations for this implementation of a priority queue?

Minute Essay

- What do you see as the relative advantages and disadvantages of the array and list based implementations of stacks and queues?
- The midterm is next class. The office hours on Monday will be used for a review session if you show up with questions for me to answer.
- Interclass Problem – Make your GUI calculator work properly. We'll look at these after the midterm.