Dynamic Stacks and Queues

11-14-2001

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the grading of assignment #4? How about the statement of #6?
- Can someone come up to the board and draw a diagram of a linked list and show me how we do an insert? How about a delete?

Implementing a Linked List

- We will now look how a linked list can be implemented in code.
- We will look at two ways of doing this. We will write part of the implementation for when the list class is a friend of the element class then look at how it could be done using recursive functions in the element class.

Stacks with Linked Lists

- You can quite easily implement a stack with a linked list. Here the head points to the top of the stack.
- Push is done by inserting an element that becomes the new head.
- Pop is done by removing that first element then doing head=head->next and returning what had been in head.

Queues with Linked Lists

- For a queue you need both the head and the tail so that you can push to the head and pop from the tail. This determines what direction the pointers need to point in the list.
- When you take something off you have to be able to get to the next thing. This means the pointers need to go from tail to head.

Minute Essay

- Did seeing the code help you to understand how you will create linked lists on your own? Do you have any questions about how to create of use linked lists? Keep in mind that you can make any class usable as a linked list by adding a next pointer and accessors for it.
- Next class we will talk about doubly linked lists.