

Doubly Linked Lists

11-16-2001

Opening Discussion

- What did we talk about last class?
- Who would like to draw the way that a linked list based stack does push and pop? How about a queue?
- When I say doubly linked list, what do you think that I mean?
- We are down to 7 regular class days left for this class.

Doubly Linked Lists

- We have talked about lists where you keep a single pointer to the "next" element. It is easy to imagine extending this to include a pointer to the "previous" element as well.
- A list with links both forward and backward is called a doubly linked list for obvious reasons.

Deleting from a DLL

- Because a node in a doubly linked list knows about both the next element and the previous one, you can do a delete when you only have one node. It has to do two links instead of one though.

```
rover->prev->next=rover->next;  
rover->next->prev=rover->prev;
```

Inserting to a DLL

- Same thing that was true for deleting is also true for inserting.
- Always link the new nodes to the old nodes first so you don't overwrite links.

```
newNode->next=rover;  
newNode->prev=rover->prev;  
rover->prev->next=newNode;  
rover->prev=newNode;
```

Code

- Let's look at how we would change our linked list code so that it creates and uses doubly linked lists instead of singly linked ones.
- What are the drawbacks of the doubly linked list? What are the advantages? Do you think the advantages are worth it?

Minute Essay

- We have now completed our coverage of linked lists and recursive data types. However, they will be reappearing in later discussions, just as loops, arrays, etc. have continually reappeared. Do you have any questions about them?
- Next class we start talking about inheritance. You should read 10.1 and 10.2 for that.
