

## Doubly Linked Lists

2-18-2002

---

---

---

---

---

---

---

## Opening Discussion

- Do you have any questions about the quiz?
- What did we talk about last class?
- Do you have any questions on assignment #2? Remember that it is due on Friday.
- Inheritance questions
  - Constructors and destructors
  - Private inheritance
  - Inheritance in Java (and other OOPLs)

---

---

---

---

---

---

---

## Deleting from a Linked List

- In our earlier discussion of linked lists we didn't really cover deleting. This isn't a straightforward process.
- When we delete we want to "link around" a node. This requires keeping track of the node before the one we want to delete.
- There are many different approaches to exactly how you implement these lists. The books is a bit different from mine.

---

---

---

---

---

---

---

## Keeping a Previous Pointer

- The real key with deleting is that you have to know the node before the one you want to delete. This basically means that as you walk the list searching for the one to remove you have another pointer that follows it, always lagging one step behind.
- This can be implemented in a single for loop.

---

---

---

---

---

---

---

---

## Doubly Linked Lists

- Another variation on the linked list is the doubly linked list. This type of list has two pointer. One to the next element and one to the previous.
- A key advantage of this is that given a node you can delete it. With the singly linked list you had to walk to that node so that when you reached it you could also have the previous one.

---

---

---

---

---

---

---

---

## More on Doubly Linked Lists

- Next time we will look at sorted lists and there being doubly linked has similar advantages.
- Of course, having more links incurs more memory overhead and also means we have to redo more links each time we add or delete.
- Just like with singly linked lists I think that "drawing pictures" can be very helpful.

---

---

---

---

---

---

---

---

## Circular Linked Lists

- Another twist that we can put on linked lists is to make them circular so that the end of the list is linked to the beginning of the list.
- This changes two things. First, there isn't a node that has no previous. Second, there isn't an end that is marked by a next of NULL. This means we have to take care in our traversals.

---

---

---

---

---

---

---

---

## Code

- Lets now go and write some code to do the things we talked about today.
- We'll start by putting in a delete in a singly linked list, then write a bit of code to implement a doubly linked list.

---

---

---

---

---

---

---

---

## Minute Essay

- Assume a circular doubly linked list class CDLL. Write the delete method for this class.
- I should be around most all of the afternoon if you have questions about anything.

---

---

---

---

---

---

---

---