Sorted Lists

2-20-2002

Opening Discussion

- What did we talk about last class? Do you have any questions about assignment #2?
- So far we have talked about linked lists where we didn't care about the order of the objects in them. How does the linked list change if we want it to be sorted?

Sorted Linked Lists

- There is a funny thing about linked lists. Given a random linked list the task of sorting it is non-trivial. (Think about trying a bubble sort. Selection/min/max sort is easier.)
- However, building a sorted linked list is quite easy as we can quickly insert into a linked list so all we have to do is find the right location and insert there.

Inserting into a Linked List

- Inserting into a linked list is very much like deleting. For a singly linked list we need to keep a previous pointer as we walk through because typically to find the location we want we have to overshoot by one.
- With a doubly linked list this isn't needed and again we can insert very efficiently if we are handed the node to insert before or after.

Code

- Now we will go back to our linked list code and create a subclass of it that is a sorted linked list.
- If we have time we will also write a doubly linked list class just so you can start to see how it works.

Minute Essay

- Your book presents an STL list class. This class has nodes that contain the objects of the type the list is created for. What limitation does this place on the use of the list? Why might you not want to do this for the list in your assignment (for this think of how many objects of that type actually exist in certain operations).
- Assignment #2 is due Friday.