

Dynamic Lists, Stacks, and Queues

11 12 2001

Opening Discussion

- What did we talk about last class?
- Because file streams aren't in the global space you have to pass them between functions as arguments (reference ones preferably). You don't want to redeclare the streams in each function.
- Imagine writing MS Word or some other program you use regularly. Would it be useful with you using file streams?

const Methods

- Methods of a class that don't change the class should be declared const. This is done by putting the const keyword after the argument list.
- All get functions should be const because they only return a value, they shouldn't change anything in the class.

```
int getLength() const;
```

Recursive Types

- A recursive function is a function that calls itself. A recursive type is a type that refers to itself.
- Recursive types can only be built with pointers because a type can't have a full copy of itself inside it. It can have a pointer to another object of that type in it.
- Today we will look at one of the simplest recursive types.

Linked Lists

- One way of creating a list is to have each element of the list know about, or link to, the next one.
- The way we create this link is with a pointer. Frequently we call it the next pointer.
- The entire list can be reached just given the head of the list.
- Linked lists can be made for any type of data by having a next pointer in a class that has other data in it.

Traversing a Linked List

- When you wanted to walk an array you had a loop that counted through the elements of the array (0..length).
- For a linked list we have a pointer that starts at the head and repeatedly move to the next element until it is NULL.

```
for(list *r=head; r!=0; r=r->next)
```

- With a class you can also traverse through recursion.

```
void List::visit() {  
    next->visit();  
}
```

Deleting from a Linked List

- To delete from a linked list all you have to do is link around it. So you take the next pointer of the one before it and make it equal to the next pointer of the node you are removing. This means that you need access to the previous one though.

```
List *prev=0,rover=head;
while((rover!=0) && (data!=dataToDelete)) {
    prev=rover; rover=rover->next;
}
if(rover!=0) {
    if(prev==0) head=rover->next;
    else prev->next=rover->next;
}
```

Inserting into a Linked List

- In a similar way, you can easily insert elements into a linked list.
- You have to be careful with the order you do it in though because you don't want to overwrite a value that you need before you have a copy of it in a new place. This means you link the new element to the rest of the list before you link the previous element to it.

Minute Essay

- What are the advantages of a linked list over an array based list? What are the disadvantages?
- You will have to think a bit about dealing with linked lists because for assignment #6 you have to do all the things we've done with array based lists using them.
- Remember that assignment #5 is due today.
