

CS180 - Lists

Note Title

10/21/2011

Announcements

- HW due Monday
- No lab next week
- Review Thurs, test Friday

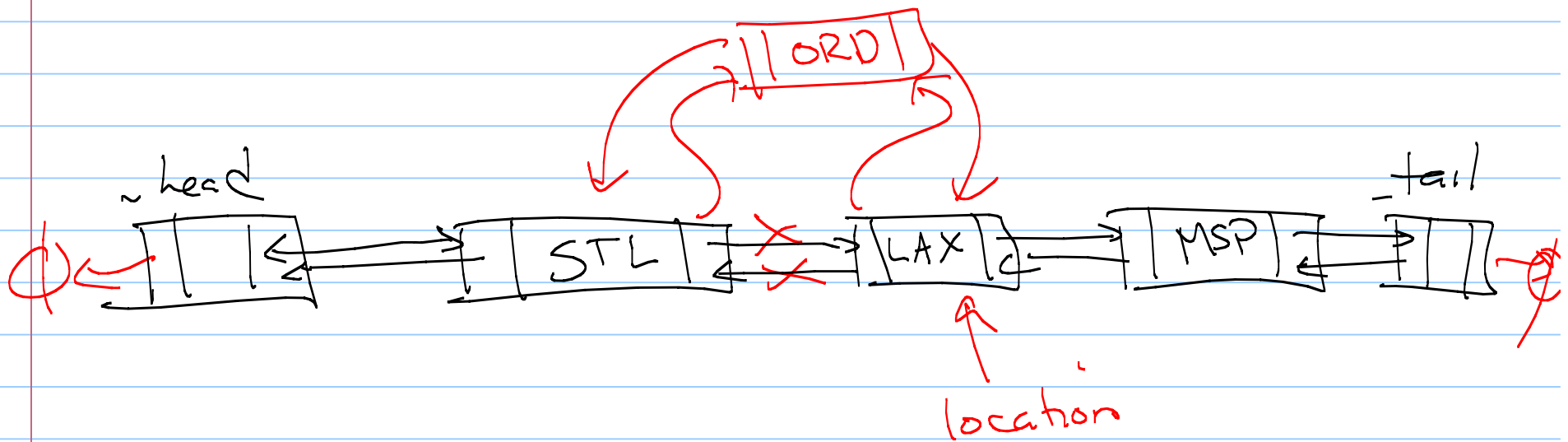
Lists:

Motivation: insert in vectors is slow!

(Running time?) $O(n)$

Idea: If I know where the element should go, inserting should be easy.

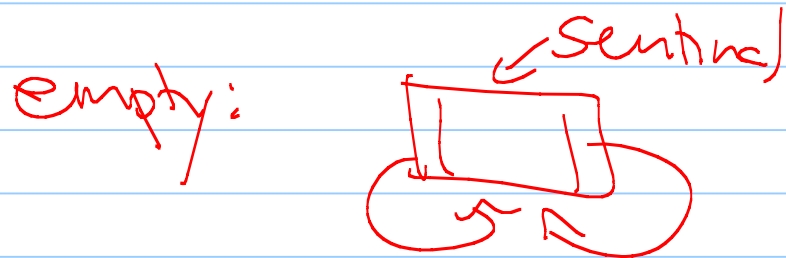
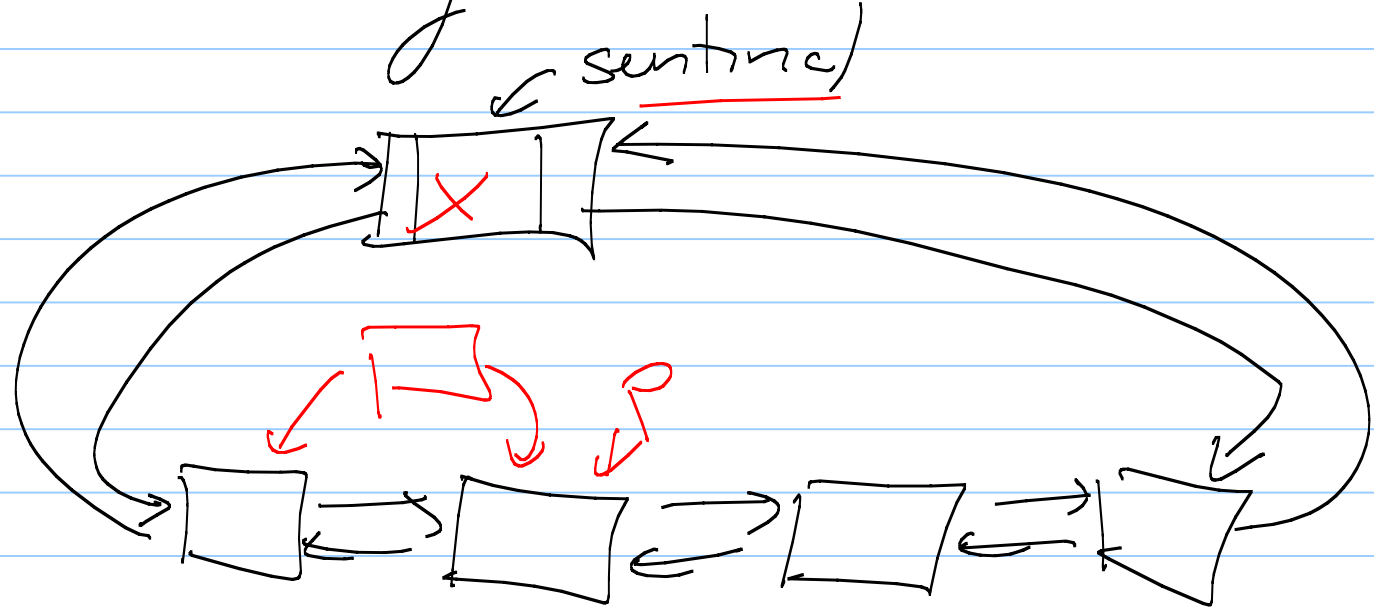
Doubly Linked Lists



insert (ORD) - where?

1 node allocation
+ 4 pointer updates

Better: Circularly linked lists



Problem: Pointers!

What do we need in order to know where we should insert?

- need pointer to a Node
private!

Solution: Iterator
class whose private data
is simply one pointer.

hide all possibility of seg
fault

Iterators

An iterator will give the user a "pointer", but with a heavily controlled structure.

(So they can't touch nodes directly.)

Compromise: Functionality versus info. encapsulation

goal: protect data & user

STL functions

Usage:

```
List<int> mylist;
```

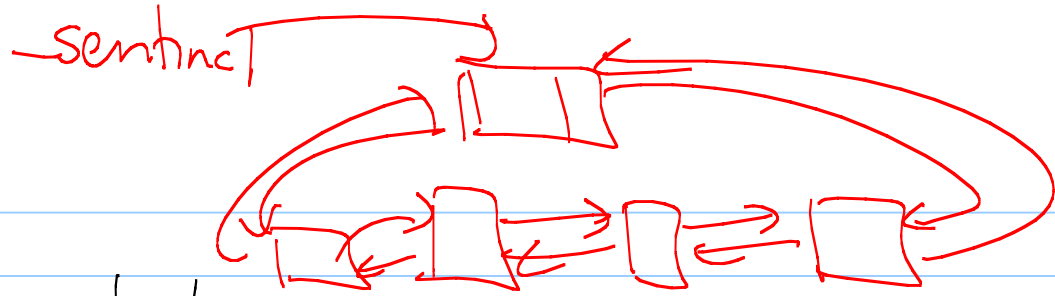
```
List<int>::iterator it;
```

```
it = mylist.begin();
```

```
it++; // now at 2nd element in list
```

```
mylist.insert(it, 5);
```

```
for (it = mylist.begin(); it != mylist.end(); it++)  
    cout << *it << " ";  
cout << endl;
```



operator++

operator*

Code:

2 internal classes

✓

$$A[i++] = \text{value};$$

$$A[++i] = \text{value};$$