

CS231 Fall 2013 Lecture 17

Stephanie R Taylor

October 11, 2013

1 Administrative Topics

- We take the quiz.

2 Recursion in linked lists

One place where recursion can be very helpful is performing operations in certain abstract data types. You will find recursion almost necessary to do anything with trees. But it also lends itself naturally to linked lists. Let's write the following two methods in our `StringLinkedList` class, using recursion.

- Write a `length` method. The basic strategy is to start at the beginning of the list, then asks the rest of the list how long it is and add 1 to that result before returning it. The base case is that we are at the end (have a null pointer), in which case we return 0.
- Write a `contains` method. The basic strategy is to return `True` if the current node contains the given item. Otherwise, ask the rest of the list if it contains the item. Return that answer.

And here is the code:

```
private int length( StringNode node ) {  
    if (node == null) {  
        return 0;  
    }  
    return 1 + length( node.next );  
}  
  
public int length() {  
    return length( this.head );  
}  
  
private boolean contains( StringNode node, String target ) {  
    if (node == null)  
        return false;  
    if (node.data.equals( target ))  
        return true;  
    return contains( node.next, target );  
}  
  
public boolean contains( String target ) {  
    return contains( this.head, target );  
}
```