

CSSE 230 Day 7

Recursion Again (and again ...)

Check out from SVN: *Recursion* and *Trees* projects

Agenda

- ▶ Recursion review
- ▶ Recursion programming exercise
- ▶ Questions?

Q1-Q3

Weiss's Recursion Principles

1. **Base Case:** Always have at least one case that can be solved without recursion.
2. **Make Progress:** Every recursive call must progress toward some base case.
3. **"You gotta believe":** Always assume that the recursive call does what it is supposed to do.
 - Use that result in building the "higher-level" solution

Q4

Recursive List Size

```
public class ListNode<T> {
    T element;
    ListNode<T> next;

    public ListNode(T e,
        ListNode<T> n) {
        this.element = e;
        this.next = n;
    }

    public ListNode(T e) {
        this(e, null);
    }

    public ListNode() {
        this(null, null);
    }
}
```

```
public class LinkedList<T> {
    private ListNode<T> header,
        private ListNode<T> last;

    // lots of other stuff.
    // Write a size() method.

}
```

Fibonacci Numbers

- ▶ Each Fibonacci number (except the first two) is the sum of the previous two Fibonacci numbers.

i	0	1	2	3	4	5	6	7	8
F_i	0	1	1	2	3	5	8	13	21

- ▶ $F_0=0$, $F_1=1$, $F_{i+2} = F_i + F_{i+1}$

```
public static int fib(int n) {
    if (n < 2)
        return n;
    return fib(n-2) + fib(n-1);
}
```

The Trouble with Fib

- » Easy to program!
- Expensive!

```
public static int fib(int n) {
    if (n < 2)
        return n;
    return fib(n-2) + fib(n-1);
}
```

Q5,Q6

Weiss's Fourth Recursion Principle

- ▶ **Compound Interest rule:** Don't recursively recompute the same things over and over in separate recursive calls.
- ▶ Alternatives:
 - Cache previously computed values
 - Use a loop
- ▶ This is a reminder from 220/221.

Q7-Q8

Recursive ParseInt?

- ▶ Input: a string representation of a positive integer
- ▶ Output: the integer
- ▶ ...using recursion

Recursive binary search is elegant

- ▶ Input: an array of integers and an element for which to search.
- ▶ Output: the index where it was found.
 - -1 if not found
- ▶ Big-Oh runtime of binary search?

Famous Diversion – Towers of Hanoi (a relevant interlude)

- ▶ The Towers of Hanoi puzzle was invented by the French mathematician Edouard Lucas in 1883.
- ▶ We are given a tower of disks initially stacked in decreasing size on one of three pegs
- ▶ The objective is to transfer the entire tower to one of the other pegs,
 - ▶ moving only one disk at a time and
 - ▶ never placing a larger disk on top of a smaller disk



Image is from

<http://www.cut-the-knot.com/recurrence/hanoi.html>

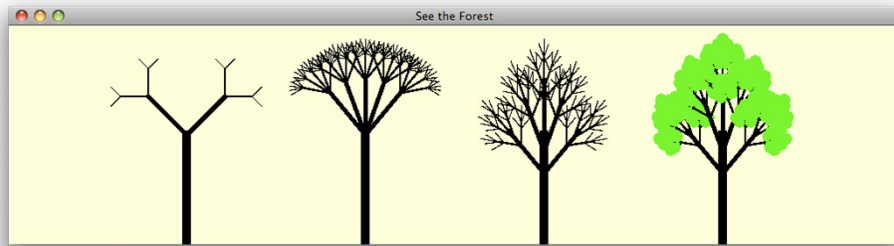
Towers of Hanoi – (my)hands on

Demo!

Towers of Hanoi

- ▶ Write the method (and its recursive helper)
- ▶ Analyze it: count the total moves required to move n disks from one peg to another
 - I.e., write and solve the recurrence relation

Trees



- ▶ Read assignment linked from schedule, WA3
- ▶ Check out **Trees** project from individual SVN repository