

## Recursion

**Numerics and Divide and Conquer**  
**3-20-2002**

---

---

---

---

---

---

---

## Opening Discussion

- What did we talk about last class?
- Do you have any questions about assignments?
- When I say divide and conquer what comes to your mind? What do you think that this means in the context of programming?

---

---

---

---

---

---

---

## Numerics

- There are a number of different numerical algorithms that can be nicely (and efficiently) expressed as recursive functions. Your book goes through a few examples.
- They are useful for doing cryptography
  - Modular exponentiation
  - Greatest Common Divisor (GCD)
  - Multiplicative inverse

---

---

---

---

---

---

---

## Greatest Common Divisor

- Euclid's algorithm for finding the GCD of two numbers is an example of an algorithm that is nicely stated in a recursive form.

```
int gcd(int a,int b) {  
    if(b==0) return a  
    return gcd(b,a%b);  
}
```

---

---

---

---

---

---

---

---

## Divide and Conquer Algorithms

- One of the standard methods of solving problems in computer science is the use of divide and conquer algorithms.
- This structure of D&C algorithms in general is to take the problem and break it into smaller pieces, then produce the solution from solutions to the smaller problems.
- Your book examines the maximum continuous subsequence problem.

---

---

---

---

---

---

---

---

## Evaluating Infix Expressions

- This is just an example I thought of last night that works here. It's not a standard example, but that's part of why I like it.
- When you write a math expression you typically do it in what is called "infix" notation. (i.e.  $(3+4)*6-8$ ) The name implies the operator is between the operands. We can evaluate this using divide and conquer.

---

---

---

---

---

---

---

---

## Approach

- At each step we want to find the lowest precedence operator and operate it on the value of the expressions to the left and right of it.
- If we ignore exponentiation all operators have higher precedence from left to right and \* and / come before + and -. Parenthesis can modify this. We do things in parenthesis after the things outside of them.

---

---

---

---

---

---

---

## Code

- Now let's code up a little recursive function that evaluates infix expressions using divide and conquer.

---

---

---

---

---

---

---

## Minute Essay

- Your book goes into detail discussing the analysis of divide and conquer algorithms. You should certainly read that section just to see some good application of analysis techniques. What order do you think the expression evaluator is?
- Remember to turn in your solution to assignment #3 ASAP.

---

---

---

---

---

---

---