Math 135 - Recursive Algs

Note Title 3/15/2010

Announcements

- HW due Wednesday in class
- Next midterm - March 31 (Wed.) in class

Recursive Algorithms

Dfn: A recursive algorithm solves a problem by reducing it to an instance of the same problem with smaller input.

Note – similar to induction!

(so don't forget a base case!)

Recall:
Recursive definition of $n!$ was:

$$n! = n \cdot (n-1)!$$

$$1! = 1$$

This leads to a recursive algorithm!

Pseudo code:

```
procedure factorial(n):
    if n = 1
        return 1
    else
        return n * factorial(n-1)
```

factorial(5) = $5 \cdot fac(4) = 5 \cdot 4 \cdot fac(3) = 5 \cdot 4 \cdot 3 \cdot fac(2)$

$= 5 \cdot 4 \cdot 3 \cdot 2 \cdot \underbrace{fac(1)}_{\text{base case}} = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$

Recall: recursive defn for $a^n = a \cdot a^{n-1}$

base case: $a^0 = 1$

Pseudocode:

```
procedure exp(a, n):
    if n = 0
        return 1
    else
        return a * exp(a, n-1)
```

1 recursive call

Computing Fibonacci Numbers

$f_n = f_{n-1} + f_{n-2}$, $f_0 = 0$, $f_1 = 1$, $(f_2 = 1)$

```
procedure fib(n):
    if n=1 or n=0
        return n
    else
        return fib(n-1) + fib(n-2)
```

↑ 2 recursive calls