Data Structures and Algorithms

Spring 2008-2009

Outline

1. Algorithm Analysis (contd.)
   - Computing Fibonacci Numbers

Outline

A Function for Fibonacci

The *n*th Fibonacci is given by $F_n = F_{n-1} + F_{n-2}, F_0 = F_1 = 1$

```
unsigned long int
fib(const unsigned int n)
{
  if (n <= 1)                       // l. 4
    return 1;                       // l. 5
  else return fib(n-1) + fib(n-2); // l. 6
}
```

- This is *not* a good example of the use of recursion
- Why not? (Write down the calls made in evaluating say, `fib(6)`; look at no. of times `fib(2)` gets called)

A Function for Fibonacci (contd.)

- Let $T(n)$ be the work done in running the function `fib()` when given the integer n
- Line 5 only gets executed when $n = 0, 1$ and only costs 2 units in those two cases
- In all other cases the cost is 1 unit for line 4 plus the cost of line 6
- Line 6 has two non-trivial function calls and an addition
- So for $n \geq 2$ cost is $1 + T(n-1) + 1 + T(n-2)$, and, in general

$$T(n) = \begin{cases} 2 & \text{for } n = 0, 1 \\ T(n-1) + T(n-2) + 2 & \text{for } n \geq 2 \end{cases}$$

A Function for Fibonacci (contd.)

- Since it is true that $f(n-1)+f(n-2) < f(n-1)+f(n-2)+2$
- $T'(n) = T'(n-1)+T'(n-2) < T(n) = T(n-1)+T(n-2)+2$
- So $T(n) = T(n-1)+T(n-2)+2 > F_{n-1} + F_{n-2} = F_n$
- Therefore, $\forall n, T(n) > F_n$, where F_n is the nth Fibonacci number, itself
- Can show (by induction) that $F_n < (\frac{5}{3})^n$
- Can also show that $F_n \geq (\frac{3}{2})^n$
- And since $T(n) > F_n$ the running time of this function is exponential and terrible