

Quiz 13: Recursion SOLUTION

CSCI 110 Section 1

Monday, September 26, 2016

- 1) Write a recursive function that takes an integer n and returns the sum of all consecutive numbers from n to 1. Hint: what's the base case? Then work through some examples. [100 points]

What should my return type be? int because we want to return a sum which is a number. float, double, or long would also have worked, but you would have to make the parameter n match the return type.

What parameters should I have? one, an integer n

```
int sum(int n) {  
    // base case: sum of ints from 1 to 1 == 1  
    if (n == 1) {  
        return 1;  
    }  
    return n + sum(n-1);  
}
```

- 2) Write a recursive function that takes an integer n and returns the value of the n th fibonacci number. The fibonacci sequence is 1, 1, 2, 3, 5, 8, 13, 21... where the next number in the sequence is calculated by taking the sum of the two previous numbers. So $\text{fib}(1) == 1$, $\text{fib}(2) == 1$, $\text{fib}(3) == 2$, $\text{fib}(4) == 3$, $\text{fib}(5) == 5$, etc. [extra credit, 100 points]

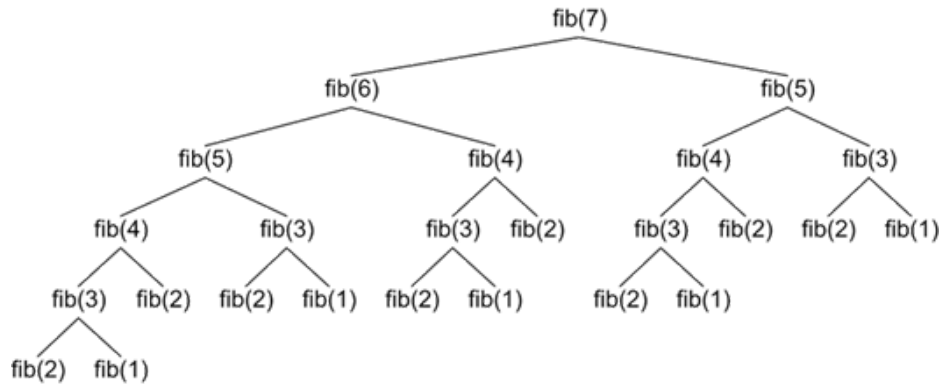
What should my return type be? int

What parameters should I have? one, an integer n

```
int fib(int n) {  
    // base case: the first two fibonacci numbers are 1  
    if (n <= 2) {  
        return 1;  
    }  
    return fib(n-1) + fib(n-2);  
}
```

- 3) Using a recursive function to calculate fibonacci numbers is extremely inefficient. Why? [extra credit, 25 points]

We do the same work over and over again. Think about what happens when you call fib(7):



We compute fib(1) five separate times. We compute fib(2) eight separate times. And so on. It turns out that the total number of recursive function calls grows exponentially. (2 to the power of n)

It's a good exercise to do to learn about recursion but it's much better to compute fibonacci numbers with a for loop:

```
int n = 7; // we want to find the seventh fibonacci number
int fib_minus_2 = 0;
int fib_minus_1 = 1;
int fib;
for (int i = 0; i < n-1; i++) {
    fib = fib_minus_1 + fib_minus_2;
    fib_minus_2 = fib_minus_1;
    fib_minus_1 = fib;
}
// the value of fib is now 13
```

Or even better, with a formula

<http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/fibFormula.html#section1>