

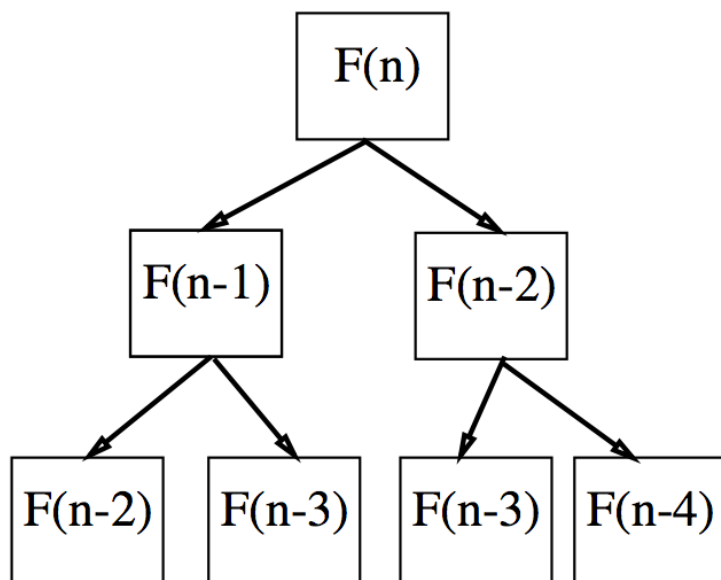
Tarea 5

Problema 5.

Esta pregunta depende del código implementado. Voy a suponer que la función implementada es la siguiente:

```
int F ( int i ) {  
    if ( i < 1 ) return 0 ;  
    if ( i == 1 ) return 1 ;  
    return F ( i - 1 ) + F ( i - 2 ) ;  
}
```

Olvidémonos del lenguaje C un poco y definamos la función $F: \mathbb{N} \rightarrow \mathbb{N}$ como función que me regresa el n -ésimo número de Fibonacci.



Sea $g: \mathbb{N} \rightarrow \mathbb{N}$ la función tal que $g(n)$ es el número de llamadas a la función F en la pila. Es fácilmente calculable. Para ello necesitamos calcular g a los primeros dos términos:

$$g(0) = 1$$

$$g(1) = 1$$

Tarea 5

Para calcular cuántas llamadas se hacen para n , g contará las llamadas a los dos números anteriores y la misma llamada:

$$g(n) = g(n-1) + g(n-2) + 1.$$

Esto es:

$$g(n) = F(n) * 3 + F(n-5).$$

Lo cual se puede aproximar para $n > 5$ por la fórmula de Binet:

$$g(n) \approx 3 * \left(\frac{\Phi_1^n - \Phi_2^{-n}}{\sqrt{5}} \right) + \frac{\Phi_1^{n-5} - \Phi_2^{-n+5}}{\sqrt{5}}.$$

Donde $\Phi_1 = \frac{1+\sqrt{5}}{2}$ y $\Phi_2 = \frac{1-\sqrt{5}}{2}$.