

Recurrencia

Recursion
Tabulación / Memoization
Dynamic Programming
(- -)

I Recursion

① Torre de Hanoi

$H(n, A, B, C)$
si $n > 0$
 $H(n-1, A, C, B)$
 move one disk de A a C
 $H(n-1, B, A, C)$

$f(n)$ = Cantidad de movimientos de $H(n, A, B, C)$?

$f(0) = \cancel{0}$

$f(1) = 1$

$f(2) = \cancel{3}$

$f(3) = 7 = f(2) + 1 + f(2) = 2 \times f(2) + 1 = 2 \times 3 + 1 = 7$

$f(n) = 2f(n-1) + 1$

① Adivinando y verificando

$f(n) = 2^n - 1$

$$= \begin{cases} n=0 & 2^0 - 1 = 1 - 1 = 0 \\ n=1 & 2^1 - 1 = 1 \\ n=2 & 3 \\ n=3 & 2^3 - 1 = 8 - 1 = 7 \end{cases}$$

$f(n-1) = 2^{n-1} - 1$

Verificamos $f(n) = 2 \times (2^{n-1} - 1) + 1$

$= 2^n - 2 + 1 = 2^n - 1$

Ejercicio en casa

Hacer el mismo análisis por n discos de tamaño i
con $i = 1 \dots n$

② Fibonacci (n)

$$F(n) = F(n-1) + F(n-2)$$

$$F(0) = 1$$

$$F(1) = 1$$

$$F(2) = 2$$

$$F(3) = 3$$

$$F(4) = 5$$

$$F(5) = 8$$

Formule explicite para $F(n)$?

Algoritmo para calcular $F(n)$!

① Solución Recursiva

Fib (n)

($n \geq 0$)

if $n < 2$ return 1
else return Fib(n-1) + Fib(n-2)

Recurrence

Tiempo de calculo?

$S(n)$ = Cantidad de $\oplus$ sumas ejecutada por Fib(n)

$$S(0) = S(1) = 0$$

$$S(2) = 1$$

$$S(3) = 2$$

$$S(4) = S(3) + S(2) + 1$$

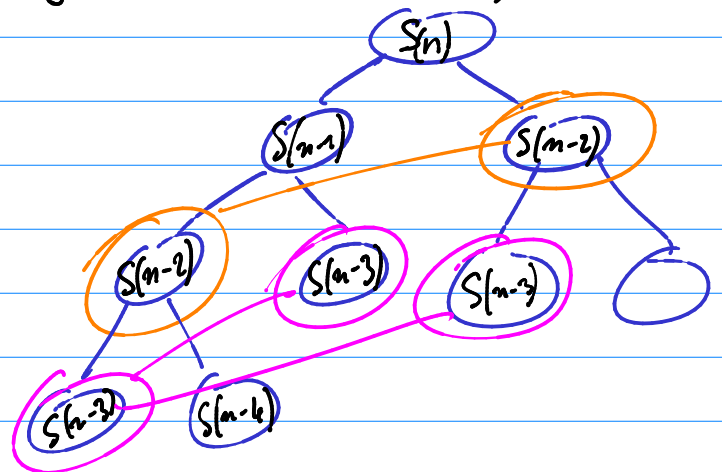
$$S(n) = S(n-1) + S(n-2) + 1$$

$$= S(n-2) + S(n-3) + 1 + S(n-2) + 1$$

$$= 2S(n-2) + S(n-3) + 2$$

$$= 2(S(n-3) + S(n-4) + 1) + S(n-3) + 2 = 3S(n-3) + 2S(n-4) + 4$$

(---) $\rightarrow$ algo exponencial en n , tipo (2^n) ??!!!



⑥ Algoritmo usando un arreglo $F[0..n]$

Fib 2 (n)

iniciar $F[0..n]$
 $F[0] = 1$
 $F[1] = 1$
Para i de 2 a n
 $F[i] = F[i-1] + F[i-2]$
retornar $F[n]$

Correcto ✓

Se demora

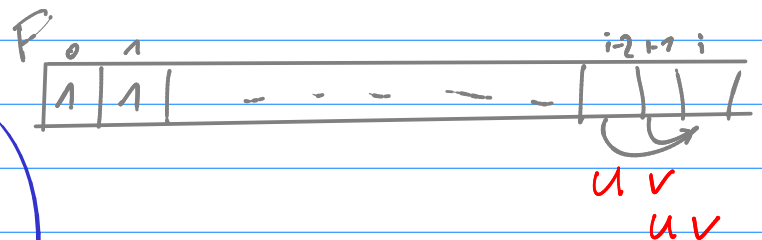
$\in O(n)$

en espacio $\in O(n)$

⑦ Algoritmo usando $O(1)$ variables

Fib 3 (n)

$U = 1$
 $V = 1$
Para i de 2 a n
 $temp \leftarrow V$
 $V \leftarrow U + V$
 $U \leftarrow temp$
retornar V



Correcto

Se demora

en espacio $O(1)$

d) Algoritmo ALGEBRAICO

$$\begin{pmatrix} F(n) \\ F(n-1) \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} F(n-1) \\ F(n-2) \\ 1 \end{pmatrix}$$

M_F

$$= M_F^2 \begin{pmatrix} F(n-2) \\ F(n-3) \\ 1 \end{pmatrix} = \dots = M_F^n \begin{pmatrix} F(0) \\ F(1) \\ 1 \end{pmatrix}$$

Se puede diagonalizar M_F ?

$$M_F = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = P D P^{-1}$$

$$M_F^n = (P D P^{-1}) (P D P^{-1}) \dots (P D P^{-1})$$

$$= P D^n P^{-1}$$

→ calcular $F(n)$ es calcular una(s) exponenciación !!!

Cuanto se demora calcular a^n ?

$$a^{\lfloor n/2 \rfloor} \times a^{\lfloor n/2 \rfloor} \times a^{n \bmod 2}$$

→ $O(\log n)$

□