Argumentos

definicion

```
def fact(n):
    if (n==0) return 1
    return n * fact(n-1)
```

Fowardl

Caso base

Llamado recursivo

O(1)

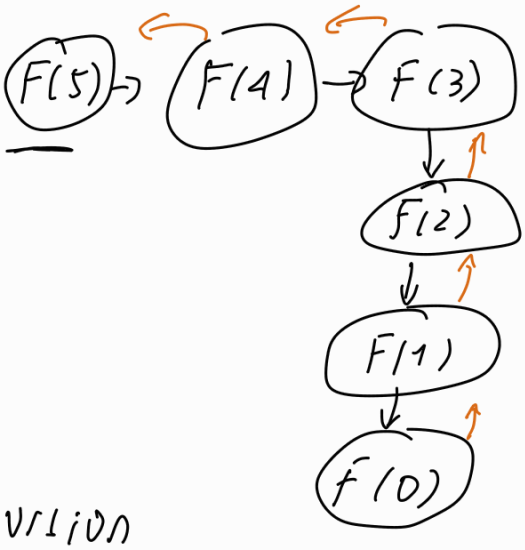
↓ Backtracking →

"Lo que se hace despues del llamado recursivo"

"Lo que se hace antes del siguiente llamado en el stack"

Stack

fact(5)



Representa un arbol

↳ Existe el árbol de recursión (AR)

↳ Representación grafica del stack y el orden de los llamados

Como calcular complejidades en una recursion

O. #nodos en AR · O(F())

6 · O(1)

Fact(n) = O(n) · O(1)

= O(n)

1. Despejar expresión matematica

→ Valor esperado.

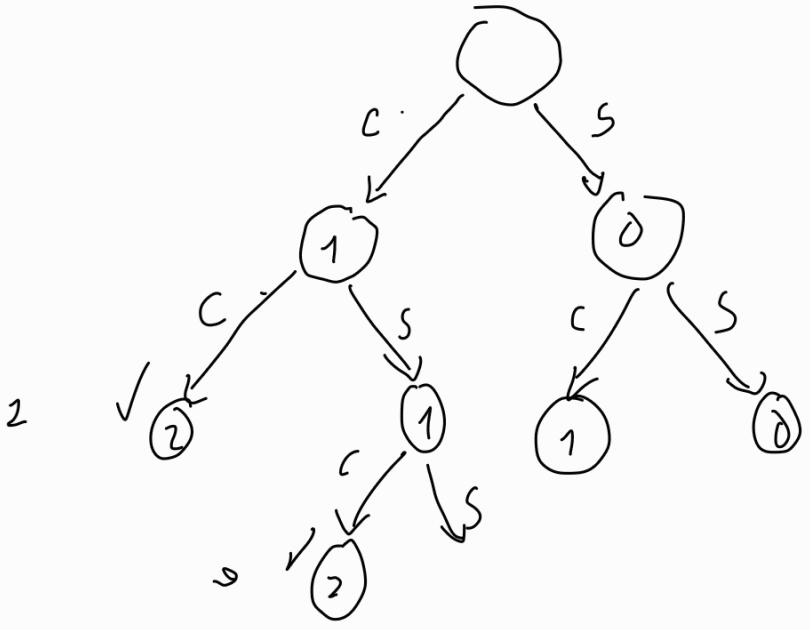
↳ # de lanzamientos de una moneda para tener 2 caras.

O(...) $X = 8x + 10 \Rightarrow x - 8x = 10$

$x(1-8) = 10 \rightarrow x = \frac{10}{7}$

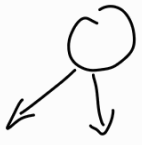
$F(n) = f(n-1) \cdot O(f)$

$\bar{x} = \bar{x}^2 + \sqrt{\bar{x}} + \text{sen}(\bar{x}) + 8$



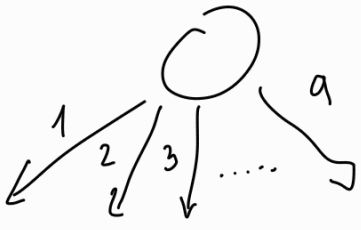
2. Arbol de recursion

Lanzamiento de moneda



Lanzamiento de un dado

Resolviendo sudoku



- Identificar comportamientos de crecimiento del arbol

óptima $\leq$ 9^9

1 nodo
- 9 arcos

+

8 nodos
- 7 arcos

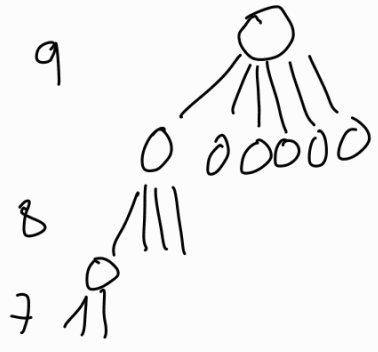
+

7 nodos
- 6 arcos

+

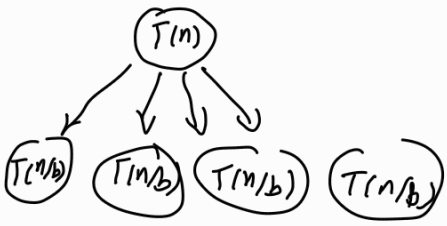
⋮

2 nodos
- 1 arco.



3. Teorema Maestro.

$$T(n) = a \cdot T\left(\frac{n}{b}\right) \cdot O(T(n))$$



Expresar una función recursiva de esta manera existen reglas (escritas) para aproximar a una expresión matemática.

$$f(a \oplus b) = f(a) \oplus f(b)$$

Divide y Venceras.

Calcular Sumas

3, 4, 5 | 8, 9, 10, | 98, 99, 100

3 + 9 + 99

39 + 297

336

- Dividir → Identificar sub problema
- Resolver → Logica de resolver un subproblema
- Unir → Como utilizar las soluciones de cada subproblema.

Caso 0 de uso de Divide y Venceras.

Paralelizar.

Multi cores.

Quad free.