Complexité, quelques relations

Dernière fois : Maths, Induction

→ Correction,

← Terminaison

Complexité

Exemples simples/ 1 appel récursif

* Factorielle :

$$C_n = C_{n-1} + \Theta(1)$$

* Tri par sélection/insert.

$$C_n = C_{n-1} + \Theta(n)$$

Notations :

* $f = O(g)$ ssi il existe $c > 0$ tq $\forall n \geq n_0, f(n) \leq c\, g(n)$
* $f = \Omega(g) \Leftrightarrow g = O(f)$
* $f = \Theta(g) \Leftrightarrow f = O(g)$ et $f = \Omega(g)$

Relations d'exemple

$$c_n = c_{n-1} + b_n$$

avec $b_n = \Theta(n^\alpha)$

Proposition Dans ce cas,

$$c_n = \Theta(n^{\alpha+1})$$

Preuve :

$$c_n = c_0 + b_1 + b_2 + \cdots + b_n$$

$$= c_0 + \sum_{k=1}^{n} k^\alpha$$

Ainsi :

Factorielle :

$$c_n = c_{n-1} + \Theta(1)$$

$$\Rightarrow c_n = \Theta(n)$$

Tri naïf :

$$c_n = c_{n-1} + \Theta(n^1)$$

$$\Rightarrow c_n = \Theta(n^2)$$

On peut avoir plusieurs appels récursifs.

$$c_n = a \times c_{n-1} + b_n$$

$$c_1 = a c_0 + b_1$$

$$c_2 = a c_1 + b_2$$
$$= a^2 c_0 + a b_1 + b_2$$

$$c_3 = a c_2 + b_3$$
$$= a^3 c_0 + a^2 b_1 + a b_2 + b_3$$

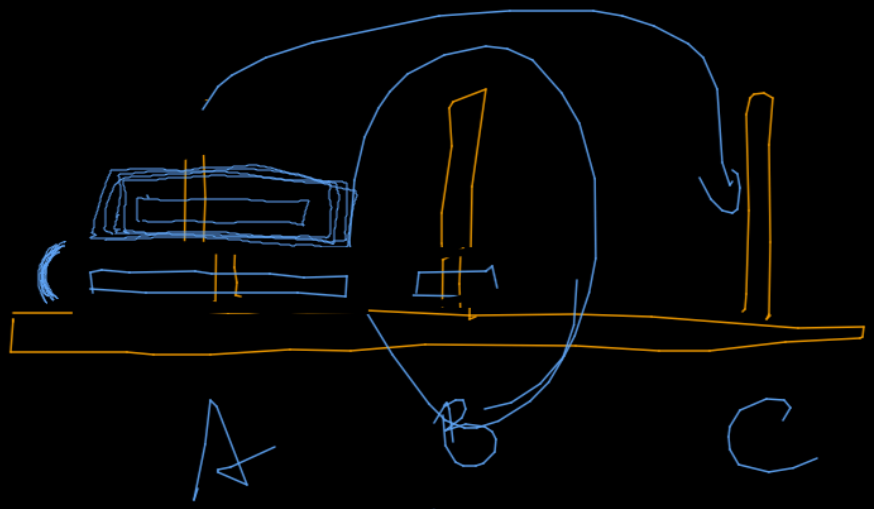
$$\frac{c_3}{a^3} = b_0 + \frac{b_1}{a} + \frac{b_2}{a^2} + \frac{b_3}{a^3}$$

$$\dots \quad \frac{c_n}{a^n} = \sum_{k=0}^{n} \frac{b_k}{a^k}$$

$$\Longrightarrow c_n = a^n \times \underbrace{\sum_{k=0}^{n} \frac{b_k}{a^k}}_{S_n}$$

* Si (S_n) majoré $c_n = \Theta(a^n)$
* $\frac{b_k}{a^k} \sim 1 \Longrightarrow c_n = \Theta(n a^n)$
* $b_n = \Theta((\lambda a)^n)$ $\lambda > 1$ $c_n = \Theta(\lambda^n)$

Tour de Hanoï



Pour résoudre avec n disques
* on mets $n-1$ disques de A vers B
* on mets disque n en C
* on mets $n-1$ disques de B vers C

$$C_n = \underbrace{C_{n-1}}_{A \to B} + 1 + \underbrace{C_{n-1}}_{B \to C}$$

$$= \underbrace{2C_{n-1}}_{a=2} + \underbrace{1}_{b_n}$$

$$\frac{C_n}{2^n} = \sum_{k=1}^{n} \frac{1}{2^k}$$

$$C_n = 2^n \sum_{k=1}^{n} \frac{1}{2^k}$$

(H) (1)

(H) (2^n)

Diviser pour Régner

Dans les exemples précédents les appels récursifs sont avec $n-1$ pour un appel avec n —

Autre idée :
on a un problème de taille n

* on coupe en sous-pbs de taille $n/2$
* on résout pour ces sous-pbs
* on recombine pour obtenir une solution du pb initial

3 phases :

* on décompose ← Diviser
* on résout les sous-pbs
* on recompose ← Régner

Exemple :

exponentiation rapide

Version naïve

$a^{n+1} = a \times a^n$

Complexité : $c_n = c_{n-1} + \Theta(1)$

$\Rightarrow c_n = \Theta(n)$

Version rapide

$$a_n = \left(a^{\left[\frac{n}{2}\right]}\right)^2 \times \begin{cases} 1 \text{ si } n \equiv 0 \,[2] \\ a \text{ si } n \equiv 1 \,[2] \end{cases}$$

Complexité

$c_n = c_{[n/2]} + \Theta(1)$

Exemple :

Buy - $ - Sell

[24; 7; 3; 12; 18; 6; 27; 1; 13]

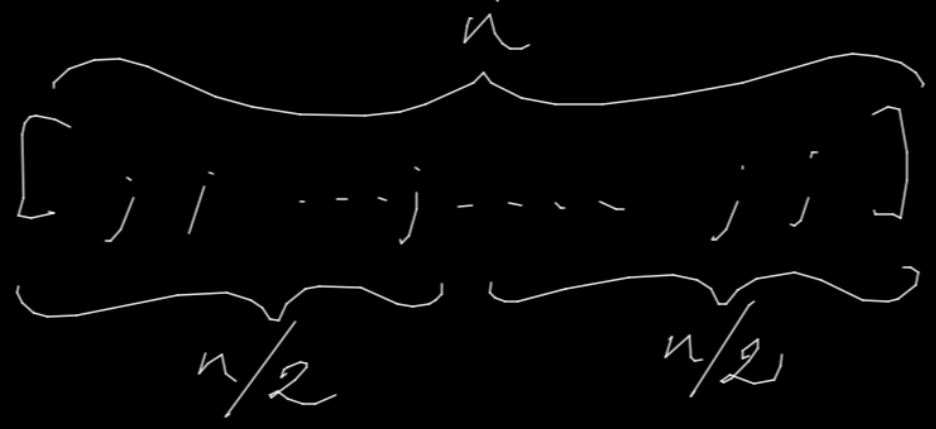
Achat ↑ Vente ↑

Version naïve simple :

① Deux boucles imbriquées

Complexité : $\Theta(n^2)$

Approche "Diviser pour Régner"



3 possibilités :
- (* A et V dans moitié 1
- (* A et V dans moitié 2
- (* A dans moitié 1 et V dans moitié 2)

Gain max. avec

(A dans moitié 1
  V dans moitié 2

→ min de moitié 1
  max " " 2

max - min

Relation de complexité :

$$C_n = C\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + C\left(\left\lceil \frac{n}{2} \right\rceil\right) + \Theta(n)$$

$\left\lfloor \frac{n}{2} \right\rfloor$: A et V dans m_1

$\left\lceil \frac{n}{2} \right\rceil$: A et V dans m_2

$\Theta(n)$: A dans m_1, V dans m_2

Solution : $C_n = \Theta(n \log_2 n)$