Diviser pour régner

(Divide et impera)

- → Diviser en sous-problèmes
- → Traiter indépendamment ces sous-problèmes récursivement
- → Fusionner les solutions des sous-problèmes

Sous-structures optimales
(pas forcément sous-problèmes répétés)

1/ Exemples (recherche dichotomique, multiplication de matrices)

2/ Analyse de complexité en étudiant les arbres de récursion

3) Théorème maître (+ sa preuve avec qq simplifications)

Exemples

A] Recherche dichotomique

Entrée : Tableau T de n éléments trié

Pb : Rechercher si un élément x est dans T (et si oui, renvoyer i t.q. $T[i]=x$)

```
function Dichotomique (T, x, début = 0, fin = n-1)
    if (début > fin)
        Return non-trouvé
    milieu = E[(début + fin)/2]
    if T[milieu] = x
        Return milieu
    if T[milieu] > x
        Return Dichotomique (T, x, début, milieu - 1)
    if T[milieu] < x
        Return Dichotomique (T, x, milieu + 1, fin)
```

$$C(n) = \begin{cases} O(1) + \max\left(C\left(\text{milieu} - \text{debut}\right), C(\text{fin} - \text{milieu})\right) & \text{si } n > 0 \\ O(1) & \text{si } n = 0 \end{cases}$$

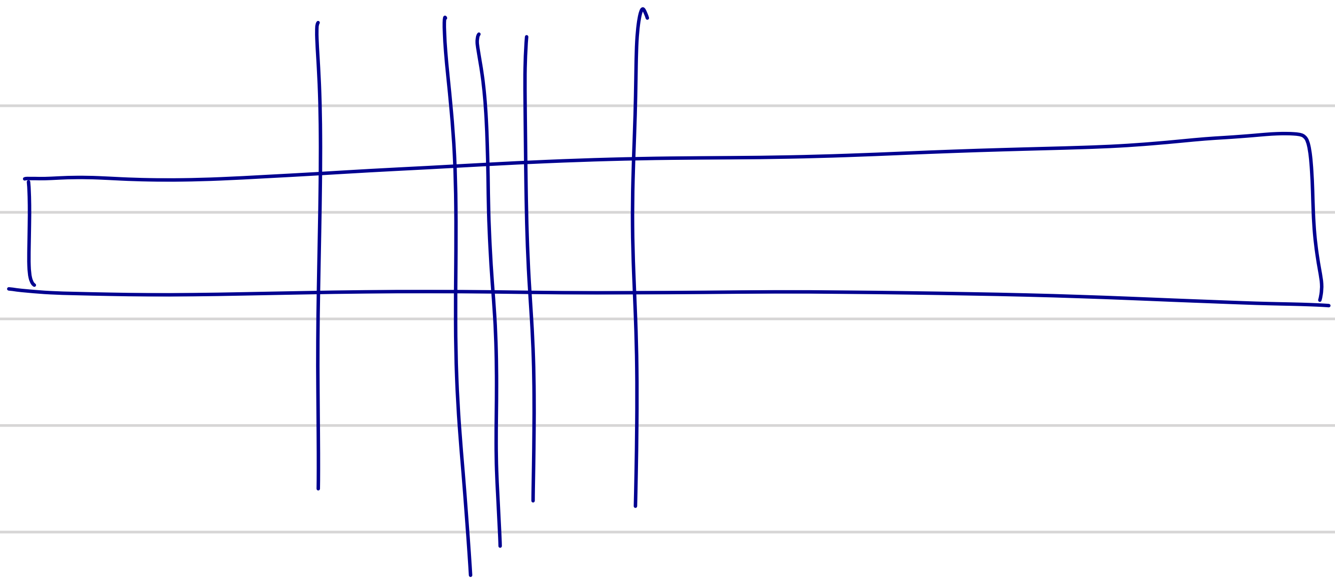
milieu - debut $\approx \frac{n}{2}$ fin - milieu $\approx \frac{n}{2}$

Suppose que n est une puissance de 2, et suppose qu'on branche en
→ début, milieu - 1) $\frac{n}{2}$
→ milieu, fin) $\frac{n}{2}$

$$\boxed{C(n) = \begin{cases} C\left(\frac{n}{2}\right) + O(1) & \text{pour } n > 0 \\ O(1) & \text{pour } n = 0 \end{cases}}$$

$\boxed{O(\log_2 n)}$

$$C(n) = C\left(\frac{n}{2}\right) + O(1) = C\left(\frac{n}{4}\right) + 2 \times O(1)$$
$$= C\left(\frac{n}{8}\right) + 3 \times O(1) = \ldots = O(1) + \log_2 n \times O(1)$$



$\alpha(n)$

$\alpha\left(\frac{n}{2}\right)$

$\alpha\left(\frac{n}{4}\right)$

$\alpha\left(\frac{n}{8}\right) \times 2$

$\times 2$

$\vdots$

$\times 2$

$\alpha(1)$

$\times 2^c$

$2^c = n \iff \log_2 n = c$

Remarque $\quad O(\log_2 n) = \underline{O(\log n)}$ pas besoin de préciser la base

car $\forall b \quad \log_b x = \dfrac{\ln x}{\ln b}$

B) Multiplication matricielle.

A, B 2 matrices de taille $n \times n$

(on suppose que n est une puissance de 2)

Calculer $C := A \times B$ $\qquad c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$

<u>Algorithme direct</u>

$\Theta(n^3)$

```
for i ← 1 to n
    for j ← 1 to n
        c_ij ← 0
        for k ← 1 to n
            c_ij = c_ij + a_ik × b_kj
```

On sait que ce problème est en $\Omega(n^2)$ car il faut calculer tous les c_{ij}.

<u>Diviser pour régner élémentaire</u>

$$A = \begin{array}{cc} & \frac{n}{2}\text{ colonnes} \\ \frac{n}{2}\text{ lignes} & \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \end{array} \qquad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

$$\begin{cases} C_{11} = A_{11} \times B_{11} + A_{12} \times B_{21} \\ C_{12} = A_{11} \times B_{12} + A_{12} \times B_{22} \\ C_{21} = A_{21} \times B_{11} + A_{22} \times B_{21} \\ C_{22} = A_{21} \times B_{12} + A_{22} \times B_{22} \end{cases}$$

$$\begin{cases} T(n) = 8 \times T\left(\frac{n}{2}\right) + \Theta(n^2) & \text{pour } n > 1 \\ T(1) = O(1) \end{cases}$$

Algorithme de Strassen

$$\begin{cases} M_1 = (A_{11} + A_{22}) \times (B_{11} + B_{22}) \\ M_2 = (A_{21} + A_{22}) \times B_{11} \\ M_3 = A_{11} \times (B_{12} - B_{22}) \\ M_4 = A_{22} \times (B_{21} - B_{11}) \\ M_5 = (A_{11} + A_{12}) \times B_{22} \\ M_6 = (A_{21} - A_{11}) \times (B_{11} + B_{12}) \\ M_7 = (A_{12} - A_{22}) \times (B_{21} + B_{22}) \end{cases}$$

$7 \times T\left(\frac{n}{2}\right)$
$+ \Theta(n^2)$

$$\begin{cases} C_{11} = M_1 + M_4 - M_5 + M_7 \\ C_{12} = M_3 + M_5 \\ C_{21} = M_2 + M_4 \\ C_{22} = M_1 - M_2 + M_3 + M_6 \end{cases}$$

$\Theta(n^2)$

$$\begin{aligned} C_{11} \overset{?}{=} & (A_{11} + A_{22}) \times (B_{11} + B_{22}) + A_{22} \times (B_{21} - B_{11}) \\ & - (A_{11} + A_{12}) \times B_{22} + (A_{12} - A_{22}) \times (B_{21} + B_{22}) \\ = & A_{11} B_{11} + \cancel{A_{11} B_{22}} + \cancel{A_{22} B_{11}} + \cancel{A_{22} B_{22}} \\ & + \cancel{A_{22} B_{21}} - \cancel{A_{22} B_{11}} - \cancel{A_{11} B_{22}} - \cancel{A_{12} B_{22}} \\ & + A_{12} B_{21} + \cancel{A_{12} B_{22}} - \cancel{A_{22} B_{21}} - \cancel{A_{22} B_{22}} \\ = & A_{11} B_{11} + A_{12} B_{21} = C_{11} \end{aligned}$$

$$\begin{cases} T(1) = O(1) \\ T(n) = 7 \times T\left(\frac{n}{2}\right) + \Theta(n^2) \end{cases}$$

Forme des récurrences

$$\begin{cases} T(1) = O(1) \\ T(n) = a\,T\left(\frac{n}{b}\right) + f(n) \end{cases}$$

2. Analyse de la complexité

$$T(n) = T\left(\frac{n}{2}\right) + O(1)$$

Dichotomique

taille n : $O(1)$

taille $\frac{n}{2}$: $O(1)$

$\vdots$

taille 1 : $O(1)$

$\log_2 n$

$$\boxed{T(n) = O(\log n)}$$

Multiplication matricielle DR élémentaire

$$T(n) = 8\,T\left(\frac{n}{2}\right) + \Theta(n^2)$$

taille n : $\Theta(n^2)$

taille $\frac{n}{2}$: $\Theta\left(\left(\frac{n}{2}\right)^2\right)$ .... $\Theta\left(\left(\frac{n}{2}\right)^2\right)$

taille $\frac{n}{4}$: $\Theta\left(\left(\frac{n}{4}\right)^2\right)$ ... $\Theta\left(\left(\frac{n}{4}\right)^2\right)$ ....

$\vdots$

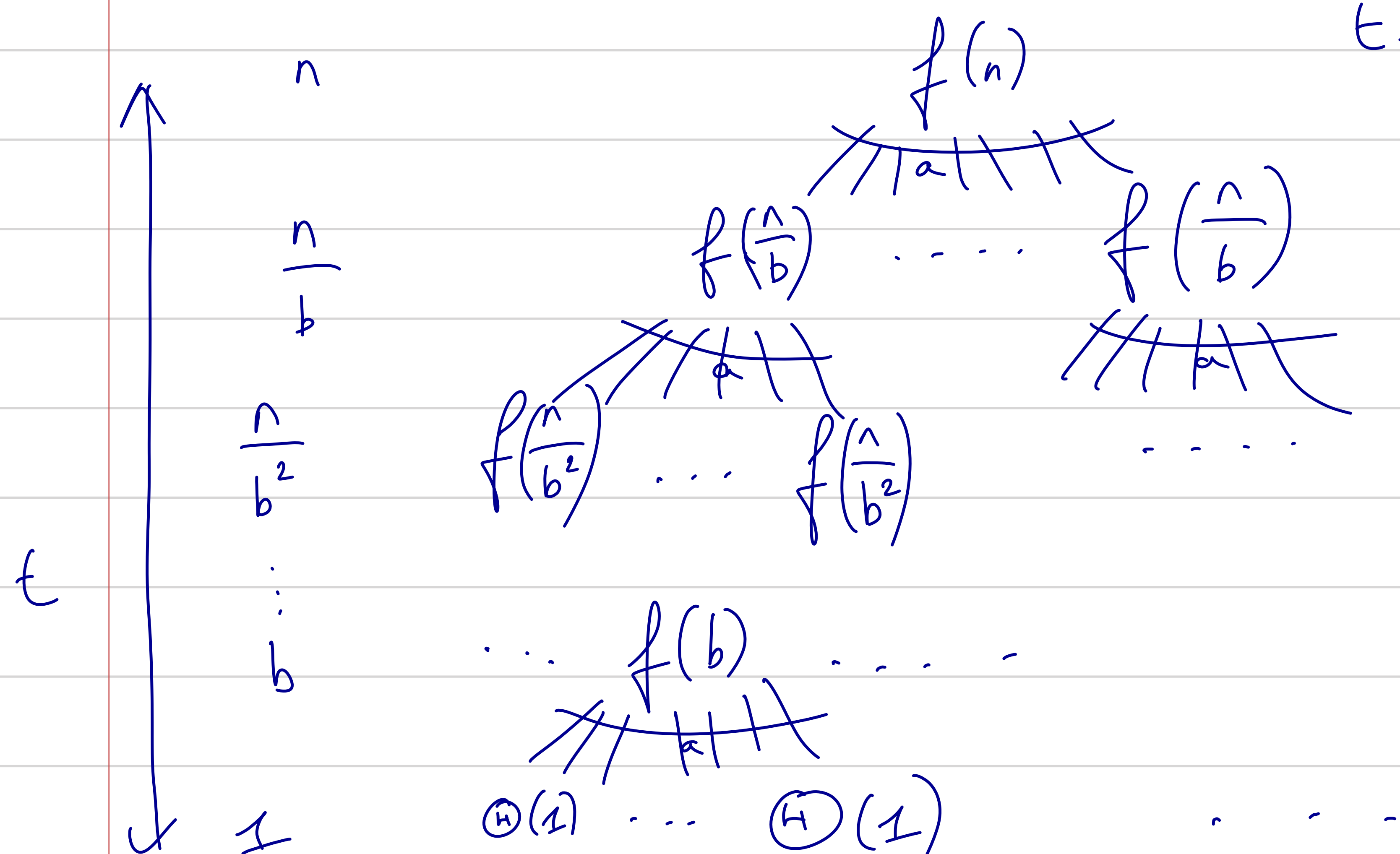
taille 2 : $\Theta(2^2)$ ..... $\Theta(2^2)$

$O(1)$ ... $O(1)$ ...

$$\begin{cases} T(n) = a\,T\left(\frac{n}{b}\right) + f(n) \\ T(1) = O(1) = \Theta(1) \end{cases}$$

(on suppose $n = b^t$)

$t = \log_b n$



$$T(n) = f(n) + a\,f\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \dots$$

$$+ a^{t-1} f(b) + a^t \times \Theta(1)$$

$$= \underbrace{\boxed{\sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)}}_{(g(n))} + \underbrace{\Theta\left(a^{\log_b n}\right)}_{\boxed{\Theta\left(n^{\log_b a}\right)}}$$

$(\Theta(n^c))$

Remarque
On a tjrs :

$$x^{\log_b y} = y^{\log_b x}$$

$$e^{\frac{\ln x \ln y}{\ln b}} \qquad e^{\frac{\ln y \ln x}{\ln b}}$$

On pose $c = \log_b a$.

3. Théorème maître

Thm : On suppose $\begin{cases} T(n) = a\,T\left(\frac{n}{b}\right) + f(n) \\ T(1) = O(1) \end{cases}$

On pose $c = \log_b a$.

calcul dominé par les feuilles de l'arbre

1er cas : Si $f(n) = O(n^{c'})$ avec $c' < c$
alors $\boxed{T(n) = \Theta(n^c)}$

calcul équilibré sur tous les noeuds de l'arbre

2ème cas : Si $f(n) = \Theta(n^c)$,
alors $\boxed{T(n) = \Theta(n^c \log n)}$.

calcul dominé par la racine de l'arbre

3ème cas : Si $f(n) = \Omega(n^{c'})$ avec $c' > c$
et si $a f\left(\frac{n}{b}\right) \leq \alpha f(n)$ pour un certain $\alpha < 1$
et n suffisamment grand, alors $\boxed{T(n) = \Theta(f(n))}$

Ne couvre pas tous les cas (par exemple si $f(n) = \Theta(n^c \log n)$)
Mais généralisable à d'autres cas.
En particulier généralisation par Akra et Bazzi quand on découpe en tailles inégales.

Dichotomie

$$T(n) = T\left(\frac{n}{2}\right) + \underbrace{O(1)}_{f(n)} \qquad a = 1 \quad b = 2$$

$$c = \log_b a = 0$$

On est dans le 2ème cas et $T(n) = \Theta(n^c \log n) = \boxed{\Theta(\log n)}$.

Mult. mat. DR élémentaire

$$T(n) = 8\,T\left(\frac{n}{2}\right) + \underbrace{\Theta(n^2)}_{f(n)} \qquad a = 8 \quad b = 2$$

$$c = \log_2 8 = 3$$

On est dans le 1er cas et $T(n) = \boxed{\Theta(n^3)}$.

Strassen

$$T(n) = 7\,T\left(\frac{n}{2}\right) + \underbrace{\Theta(n^2)}_{f(n)} \qquad a = 7 \quad b = 2$$

$$c = \log_2 7 \simeq 2{,}807$$

On est dans le 1er cas et $T(n) = \Theta(n^c)$
$= \boxed{\Theta(n^{\log_2 7})}$.

Produits matriciels

1990 Coppersmith – Winograd $n^{\approx 2{,}375477}$

2010 $n^{\approx 2{,}374}$

2011 $n^{\approx 2{,}3728642}$

2014 $n^{\approx 2{,}3728639}$

Preuve du théorème maître (dans le cas où n est une puissance de b)

$$T(n) = \underbrace{\sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)}_{g(n)} + \Theta(n^c)$$

$(c = \log_b a)$

1er cas On suppose $f(n) = O(n^{c'})$ avec $c' < c$.

On montre que $T(n) = \Theta(n^c)$.

On pose $c = c' + \varepsilon$ avec $\varepsilon > 0$.

$c = \log_b a \iff a = b^c$

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j O\left(\left(\frac{n}{b^j}\right)^{c'}\right)$$

$$= O\left(\sum_{j=0}^{\log_b n - 1} a^j \frac{n^{c-\varepsilon}}{(b^j)^{c-\varepsilon}}\right)$$

$$= O\left(n^{c-\varepsilon} \times \sum_{j=0}^{\log_b n - 1} \underbrace{\left(\frac{a}{b^c}\right)^j}_{=1} b^{\varepsilon j}\right)$$

$$= O\left(n^{c-\varepsilon} \times \sum_{j=0}^{\log_b n - 1} (b^\varepsilon)^j\right)$$

$$\boxed{\sum_{k=i}^{j} \alpha^k = \frac{\alpha^{j+1} - \alpha^i}{\alpha - 1}}$$

$$= O\left(n^{c-\varepsilon} \times \frac{(b^\varepsilon)^{\log_b n} - 1}{b^\varepsilon - 1}\right)$$

$$= O\left(n^{c-\varepsilon} \times \left[(b^\varepsilon)^{\log_b n} - 1\right]\right)$$

$$\boxed{x^{\log_\beta y} = y^{\log_\beta x}}$$

$$= O\left(n^{c-\varepsilon}\left(n^{\log_b b^\varepsilon} - 1\right)\right)$$

$$= O\left(n^{c-\varepsilon}\left(n^\varepsilon - 1\right)\right)$$

$$= O(n^c).$$

D'où $T(n) = O(n^c) + \Theta(n^c) = \Theta(n^c)$

$2^{\text{ème}}$ cas

$f(n) = \Theta(n^c)$. MQ $T(n) = \Theta(n^c \log n)$

$$T(n) = g(n) + \Theta(n^c)$$

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)$$

$$= \Theta\left(\sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^c\right)$$

$$= \Theta\left(\sum_{j=0}^{\log_b n - 1} \underbrace{\left(\frac{a}{b^c}\right)}_{=1}^j \times n^c\right)$$

$$= \Theta\left(\log_b n \times n^c\right) = \Theta(n^c \log n)$$

$3^{\text{ème}}$ cas

$f(n) = \Omega(n^{c'})$ avec $c' > c$

Et pour n suffisamment grand $a f\left(\frac{n}{b}\right) \leq \alpha f(n)$ pour un certain $\alpha < 1$.

$$f(n) \geq \frac{a}{\alpha} f\left(\frac{n}{b}\right) \geq \frac{a^2}{\alpha^2} f\left(\frac{n}{b^2}\right) \geq \dots \geq \frac{a^k}{\alpha^k} f\left(\frac{n}{b^k}\right)$$

pour $\frac{n}{b^{k-1}}$ « suffisamment grand », c-à-d. $\frac{n}{b^{k-1}} \geq \underbrace{N}_{\text{constante}}$

On suppose $n \geq N$.

On note k_0 le plus grand $k \geq 1$ t.q. $\frac{n}{f^{k-1}} \geq N$

Pour $k > k_0$, $\frac{n}{f^k} < \underbrace{N}_{\text{constante}}$ d'où $f\left(\frac{n}{b^k}\right) = O(1)$

$$g(n) = \sum_{k=0}^{\log_b n - 1} a^k f\left(\frac{n}{b^k}\right) = \sum_{k=0}^{k_0} a^k f\left(\frac{n}{b^k}\right) + \sum_{k=k_0+1}^{\log_b n - 1} a^k f\left(\frac{n}{b^k}\right)$$

$$\leq \sum_{k=0}^{k_0} \alpha^k f(n) + \sum_{k=k_0+1}^{\log_b n - 1} a^k \times O(1)$$

$$g(n) \leq \sum_{k=0}^{k_0} \alpha^k f(n) + \sum_{k=k_0+1}^{\log_b n - 1} a^k \times O(1)$$

$$\leq \left(\sum_{k=0}^{\infty} \alpha^k\right) \times f(n) + O(\log n) \times \frac{a^{\log_b n}}{a} \times O(1)$$

$$\leq \frac{1}{1-\alpha} f(n) + O\left(n^c \log n\right) = O(f(n)) + O(n^c \log n)$$

Or $f(n) = \Omega\left(n^{c'}\right)$ avec $c' > c$

En particulier, $f(n) = \Omega\left(n^c \log n\right)$

D'où $g(n) = O(f(n))$.

$$T(n) = \underbrace{g(n)}_{O(f(n))} + \underbrace{\Theta(n^c)}_{O(f(n))}$$

Et donc $T(n) = O(f(n))$.

Et $g(n) = f(n) + \ldots \geq f(n)$ d'où $g(n) = \Omega(f(n))$

Donc $T(n) = \Omega(f(n))$ et $T(n) = \Theta(f(n))$.