Diviser pour régner

(Divide et impera)

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

Sous-structures optimales

1/ Exemples (recherche dichotomique, multiplication de matrices)

2/ Analyse de la complexité en étudiant les arbres de récursion

3/ Théorème maître

Exemples

1/ Recherche dichotomique

Entrée : tableau T de n éléments trié

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

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

taille du tableau
fin - debut + 1

$$T(n) = \begin{cases} O(1) + \max\left(T\left(\text{mid} - \text{debut}\right), T\left(n - \text{mid} - 1 + \text{debut}\right)\right) & \text{si } n \neq 0 \\ O(1) & \text{si } n = 0 \end{cases}$$

$$\text{mid} = E\left[\frac{\text{début} + \text{fin}}{2}\right]$$

Supposons que n est une puissance de 2, et supposons qu'on branche en → début, mid - 1
→ mid, fin

$$\boxed{T(n) = \begin{cases} O(1) + T\left(\frac{n}{2}\right) & \text{si } n \neq 0 \\ O(1) & \text{si } n = 0 \end{cases}}$$

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

Multiplication matricielle

A, B deux matrices de taille $n \times n$
(on suppose que n est une puissance de 2)

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

Algorithme direct :

$\Theta(n^3)$

```
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 qu'une borne inférieure sur la complexité du produit matriciel est $\Omega(n^2)$.

Diviser pour régner élémentaire

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \qquad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

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

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

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

Algorithme de Strassen

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

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

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

$$\Theta(n^2)$$

$$C_{11} \overset{?}{=} (A_{11} + A_{22})(B_{11} + B_{22}) + A_{22}(B_{21} - B_{11}) - (A_{11} + A_{12}) B_{22} + (A_{12} - A_{22})(B_{21} + B_{22})$$

$$= A_{11}B_{11} + \cancel{A_{11}B_{22}} + \cancel{A_{22}B_{11}} + \cancel{A_{22}B_{22}}$$

$$+ \cancel{A_{22}B_{21}} - \cancel{A_{22}B_{11}} - \cancel{A_{11}B_{22}} - \cancel{A_{12}B_{22}}$$

$$+ A_{12}B_{21} + \cancel{A_{12}B_{22}} - \cancel{A_{22}B_{21}} - \cancel{A_{22}B_{22}}$$

$$= A_{11}B_{11} + A_{12}B_{21} = C_{11}$$

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

Forme des récurrences :

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

Dichotomique

2. Analyse de la complexité.

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

$$\left.\begin{array}{ll}
\text{taille } n: & O(1) \\
\text{taille } \frac{n}{2} & O(1) \\
\text{taille } \frac{n}{4} & O(1) \\
 & \vdots \\
\text{taille } 1 & O(1)
\end{array}\right\} \log_2 n$$

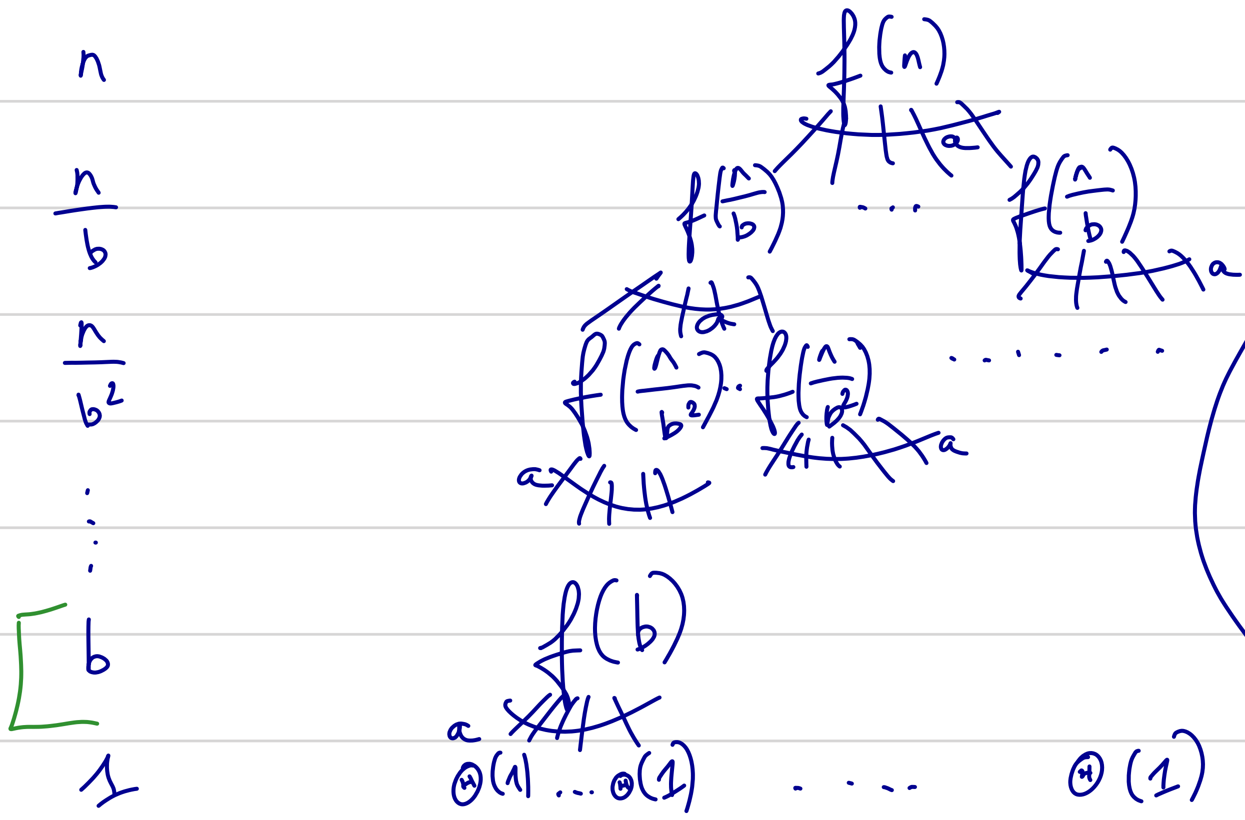
$$T(n) = O(\log n)$$

Multiplication matricielle diviser pour régner directe

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

$$\begin{array}{lccc}
\text{taille } n: & & \Theta(n^2) & \\
\frac{n}{2}: & \Theta\left(\left(\frac{n}{2}\right)^2\right) & & \Theta\left(\left(\frac{n}{2}\right)^2\right) \\
\frac{n}{4}: & \Theta\left(\left(\frac{n}{4}\right)^2\right) & & \Theta\left(\left(\frac{n}{4}\right)^2\right) \\
\vdots & & & \\
2 & \ldots\ \Theta(2^2)\ \ldots & & \ldots\ \Theta(2^2)\ \ldots \\
1 & O(1)\ O(1)\ O(1) & &
\end{array}$$

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



$$n = b^t$$

$$b = \frac{n}{b^{\log_b n - 1}} = \frac{b^t}{b^{t-1}} = b$$

$$T(n) = f(n) + a f\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \dots + \underbrace{a^{\log_b n - 1} f\left(\frac{n}{b^{\log_b n - 1}}\right)} + a^{\log_b n} \times \Theta(1)$$

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

$$\boxed{a^{\log_b n} = n^{\log_b a}}$$

$$x^y = e^{y \ln x} \qquad x = a,\quad y = \log_b n$$

$$= e^{\frac{\ln n}{\ln b} \times \ln a}$$

$$= e^{\ln n \times \log_b a}$$

$$= n^{\log_b a}$$

On pose : $c = \log_b a$.

3. Théorème maître

Thm : On suppose $T(n) = aT\left(\frac{n}{b}\right) + f(n)$ et $T(1) = O(1)$. On pose $c = \log_b a$.

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

2ème cas : Si $f(n) = \Theta(n^c)$.
Alors $T(n) = \Theta(n^c \log n)$

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

Généralisation par : Akra-Bazzi (quand on découpe en taille inégale)

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

$a = 1$
$b = 2$
$c = \log_2 1 = 0$
$f(n) = O(1)$

$$T(n) = \Theta(\log n)$$

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

$a = 8$
$b = 2$
$c = \log_2 8 = 3$

$f(n) = \Theta(n^2)$

$$T(n) = \Theta(n^3)$$

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

$a = 7$

$b = 2$

$c = \log_2 7 \approx 2,807$

$f(n) = \Theta(n^2)$

$T(n) = \Theta\left(n^{\log_2 7}\right)$ d'après le th. maître

Produits matriciels
$$\begin{cases} \text{Coppersmith-Winograd (1990)} \\ O\left(n^{\approx 2,375477}\right) \\ 2,374 \quad (2010) \\ 2,3728642 \quad (2011) \\ 2,3728639 \quad (2014) \end{cases}$$

Preuve du théorème maître

$(c = \log_b a)$

$\boxed{b^c = a}$

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

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

On montre que $\boxed{T(n) = \Theta(n^c)}$

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

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

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

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

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

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

$$= O\left[n^{c'+\varepsilon}\right] = O\left[n^c\right]$$

$2^{\text{ème}}$ cas

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

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

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

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

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

$$= \Theta\left[n^c \times \log_b n\right].$$

$3^{\text{ème}}$ cas

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

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

$$a^j f\left(\frac{n}{b^j}\right) \leq a^{j-1} \alpha f\left(\frac{n}{b^{j-1}}\right)$$

$$\leq a^{j-2} \alpha^2 f\left(\frac{n}{b^{j-2}}\right)$$

$$\leq \dots$$

$$\leq \alpha^j f(n)$$

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

$$\leq \left[\sum_{j=0}^{\infty} \alpha^j\right] f(n)$$

$$\left[\sum_{j=0}^{\infty} \alpha^j\right] f(n) = \left(\frac{1}{1-\alpha}\right) f(n) = \Omega\left(n^{c'}\right)$$

(avec $c' > c$).

D'où $g(n) = \Omega\left(n^{c'}\right)$

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

$$= O\left(g(n)\right) = O\left(f(n)\right).$$