# Mathématiques discrètes : Éléments d'asymptotique

Licence — Université Lille 1
Pour toutes remarques : Alexandre.Sedoglavic@univ-lille1.fr

Semestre 3 — 2008-09

Pour évaluer une complexité, on cherche à trouver un ordre de grandeur approximatif du comportement d'une fonction $f$ dépendente de $n$ dans des conditions limites ($n$ tend vers l'infini, zéro, etc).

Typiquement, on s'intéresse au nombre d'opérations nécessaire à un algorithme selon la taille $n$ des entrées.

Attention ces estimations sont vraies au voisinage de l'infini et ne sont donc pas toujours pertinente dans notre monde. ($\log n$ croit plus lentement que $n^{1/10000}$ mais
pour $n := 10^{100}$, on a $\log n > 230$ et $10^{1/100} < 3/2$.

Nous allons nous donner dans la suite les outils permettant de comparer le comportement à l'infini des fonctions ($n^{\log n}$ et $(\log n)^n$ par example).

# Majoration et minoration asymptotique



## Definition

Soient $f$ et $g$ deux fonctions de $\mathbb{N}$ dans $\mathbb{R}^+$.

- On dit que $g$ est un *majorant* de $f$ et on note $f \in O(g)$ si, et seulement si,

$$\exists\, c \in \mathbb{R}^+,\ \exists\, n_0 \in \mathbb{N},\ \forall n \in \mathbb{N},\ n > n_0 \Rightarrow f(n) \leq cg(n).$$

- On dit que $g$ est un *minorant* de $f$ et on note $f \in \Omega(g)$ si, et seulement si,

$$\exists\, c \in \mathbb{R}^+,\ \exists\, n_0 \in \mathbb{N},\ \forall n \in \mathbb{N},\ n > n_0 \Rightarrow f(n) \geq cg(n).$$

- On dit que $f$ et $g$ sont du *même ordre* et on écrit $f \in \theta(g)$ si, et seulement si, $f \in O(g)$ et $g \in O(f)$.

Ainsi, la fonction $n^2 + 2n$ est dans $O(n^2)$ et dans $O(n^3)$ (en fait $O(n^2) \subset O(n^3)$). On note parfois $n^2 + 2n = O(n^2)$.

# Asymptotique des équations de partitions



## Proposition

*Considérons la suite $u_n$ définie par la relation de récurrence*

$$u_n = bu_{n/a} + cn^k,\ a \geq 2,\ b > 0,\ c > 0,\ k \in \mathbb{N}.$$

*Si $u_n$ est monotone croissante à partir d'un certain rang, alors :*

$$u_n \in \begin{cases} \theta(n^k) & \text{si} \quad b < a^k, \\ \theta(n^k \log n) & \text{si} \quad b = a^k, \\ \theta\left(n^{\frac{\log b}{\log a}}\right) & \text{si} \quad b > a^k. \end{cases}$$

# Règles de calcul



## Proposition

*Pour toute constante $\alpha$ et pour toutes fonctions $g, g_1$ et $g_2$, on a en notation abrégée :*

$$\begin{array}{rcl}
\alpha O(g(n)) & \subset & O(g(n)), \\
O(O(g(n))) & \subset & O(g(n)) \\
O(g(n)) + O(g(n)) & \subset & O(g(n)), \\
O(g_1(n)) + O(g_2(n)) & \subset & O(\max(g_1(n), g_2(n))), \\
O(g_1(n)) \cdot O(g_2(n)) & \subset & O(g_1(n) g_2(n)), \\
O(g_1(n) g_2(n)) & \subset & O(g_2(n)) g_1(n).
\end{array}$$

(la première propriété signifie que pour toute fonction $f$ dans $O(g(n))$ et toute constante $\alpha$, la fonction $\alpha f$ est aussi dans $O(g(n))$.)

## Definition

Soient $f$ et $g$ deux fonctions de $\mathbb{N}$ dans $\mathbb{R}^+$.

- On dit que $f$ est *négligeable* devant $g$ et on note $f \in o(g)$ si, et seulement si,

$$\forall \epsilon \in \mathbb{R}^+,\ \exists\, n_0 \in \mathbb{N},\ \forall n \in \mathbb{N},\ n > n_0 \Rightarrow f(n) \leq \epsilon g(n).$$

## Proposition

- *Si* $f(n) \in o(1)$ *alors* $\log(1 + O(f(n))) = O(f(n))$.
- *Si* $f(n) \in O(1)$ *alors* $\exp(O(f(n))) = 1 + O(f(n))$.
- *Si* $f(n) \in o(1)$ *et* $f(n)g(n) \in O(1)$ *alors* $(1 + O(f(n)))^{O(g(n))} = 1 + O(g(n)f(n))$.

## Proposition

*Soient $f$ et $g$ deux fonctions positives ne s'annulant pas pour $n$ suffisamment grand, alors :*

1. $\lim_{n\to\infty} f(n)/g(n) = \alpha \neq 0 \Rightarrow f \in \theta(g)$,
2. $\lim_{n\to\infty} f(n)/g(n) = 0 \Rightarrow f \in O(g)$ *et* $f \notin \theta(g)$,
3. $\lim_{n\to\infty} f(n)/g(n) = \infty \Rightarrow g \in O(f)$ *et* $g \notin \theta(f)$,
4. $\lim_{n\to\infty} f(n)/g(n) = 0 \Rightarrow f \in o(g)$.

## Definition

Si $\lim_{n\to\infty} f(n)/g(n) = 1$, alors $f$ et $g$ sont *asymptotiquement équivalentes* et on note $f \sim g$.

# Règle de l'Hospital

Si la limite de $f(n)/g(n)$ quand $n$ tends vers l'infini est indéterminée, alors on peut utiliser le résultat suivant :

## Proposition

*Si la fonction $g'(n)$ (′ désigne la dérivation) ne s'annule pas pour $n$ suffisamment grand et si*

$$\lim_{n\to\infty} \frac{f'(n)}{g'(n)} = a$$

*existe ($a$ peut être infini), alors*

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = a.$$

Ainsi, on peut par exemple déterminer :

$$\lim_{n\to\infty} \frac{n^2 sin(1/n)}{3n+1}, \quad \lim_{n\to\infty} \frac{n^\alpha}{\log n}.$$

# Échelle de comparaison



## Definition

Une *échelle de comparaison* est un ensemble de fonctions : $E := \{g : \mathbb{N} \to \mathbb{R}\}$ telles que :

1. pour tout $g$ dans $E$, soit $g$ est la fonction constante 1, soit elle tends vers 0 ou l'infini quand $n$ tends vers l'infini ;
2. pour tout $g_1$ et $g_2$ dans $E$, si $g_1$ différent de $g_2$ alors $g_1 \in o(g_2)$ ou $g_2 \in o(g_1)$ ;
3. toute fonction $g$ dans $E$ a son image dans $\mathbb{R}^+$.

Par exemple, $\{n^a \log^b(n),\ (a,b) \in \mathbb{R}^2\}$ est une échelle de comparaison.

$$n^{a_1}(\log n)^{a_2}(\log\log n)^{a_3} \in o\left(n^{b_1}(\log n)^{b_2}(\log\log n)^{b_3}\right)$$
$$\Leftrightarrow$$
$$(a_1, a_2, a_3) \leq_{\text{lexord}} (b_1, b_2, b_3).$$

## Definition

S'il existe $\alpha_1$ dans $\mathbb{R}^\star$ et $\alpha_1 g_1$ dans $E$ tels que $f \sim g$, alors on dit que $g_1$ est la *partie principale* de $f$ et on note $f = \alpha_1 g_1 + o(g_1)$.

Le *développement asymptotique* d'ordre $k$ de $f$ par rapport à $E$ est une expression de la forme $f \in \alpha_1 g_1 + \cdots + \alpha_k g_k + o(g_k)$ avec $g_{i+1} = o(g_i)$.

On peut citer par exemple la formule de Stirling :

$$n! \in \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1 + \frac{1}{12n} + \frac{1}{288n^2} - \frac{139}{51840n^3} + O\left(\frac{1}{n^3}\right)\right).$$

la complexité du quicksort est de $O(n\log(n))$ en moyenne et — comme celle des autres tris — de $O(n^2)$ dans le pire des cas.

On dit qu'une fonction $f(n)$ est de complexité polynomiale si $f(n)$ est dans $O(n^k)$ pour un entier $k$.

La classe de problème $P$ est constituée de tous les problèmes qu'une machine de Turing déterministe peut résoudre algorithmiquement en un nombre d'opérations polynomiale en la taille de l'entrée.

La classe de problème $NP$ est constituée de tous les problèmes dont le résultat peut être vérifié par un processus de la classe $P$.

La question de savoir si l'on peut calculer tout ce que l'on peut vérifier $(P = NP)$ reste ouverte.