ALGORITHMIQUE : TRIS NAÏFS

INTRODUCTION DES NOTATIONS

Objectifs. — On se fixe les objectifs suivant pour ce cours

— compter le nombre d'opérations effectué par des algorithmes ;

— créer des programmes conçus pour toutes les valeurs d'un certain type.

L'évaluation de complexité est *asymptotique*. Le nombre d'opérations est exprimé en fonction de la « taille » [(1§)] des entrées.

> DÉFINITION 0.0.0.1 (Notation $\mathrm{O}()$). —
> Soient $f, g : \mathbf{N} \to \mathbf{R}^+$. On dit que $f = \mathrm{O}(g)$ si :
> $$\exists C \in \mathbf{R}^+, \exists n_0 \in \mathbf{N}, \forall n \geq n_0,\ f(n) \leq C \cdot g(n).$$

Dans la suite, $f : \mathbf{N} \to \mathbf{R}^+$ sera l'application qui à une entrée de taille n associe le nombre d'opérations nécessaires.

> DÉFINITION 0.0.0.2 (Notation $\Omega()$). —
> Soient $f, g : \mathbf{N} \to \mathbf{R}^+$. On dit que $f = \Omega(g)$ si :
> $$\exists C \in \mathbf{R}^+, \exists n_0 \in \mathbf{N}, \forall n \geq n_0,\ f(n) \geq Cg(n).$$

> DÉFINITION 0.0.0.3 (Notation $\Theta()$). —
> Soient $f, g : \mathbf{N} \to \mathbf{R}^+$. On dit que $f = \Theta(g)$ si :
> $$f = \mathrm{O}(g) \text{ et } f = \Omega(g).$$

1§. Par exemple : pour un entier $x \in \mathbf{N}$ ça peut être la longueur de son expression en décimal ; pour une liste cela peut être le nombre d'éléments de la liste, si l est de longueur n alors sa taille peut être : $\text{taille}(l) = |l| = \sum_{i=0}^{n-1} \text{taille}(l[i])$

Exercice. — Montrer que $f = \mathrm{O}(g)$ lorsque :

1. $f(n) = n^d$ et $g(n) = n^{d+i}, i \geq 0$;
2. $f(n) = \log n$ et $g(n) = n^\varepsilon,\ \varepsilon > 0$;
3. $f(n) = n^k$ et $g(n) = r^n$ avec $r > 1$ et $k \in \mathbf{N}$.

Définition 0.0.0.4 (Opérations dans les algorithmes). —
Voici les éléments comptés :
— le nombre d'affectations ;
— le nombre de comparaisons ;
— le nombre de divisions :
— ...

Exemple. — Le nombre de divisions effectuées par l'algorithme d'Euclide pour a, b (avec $a > b$) est une fonction de b : $\Theta(\log b)$.

1. TRIS

Objectifs. — Faire un tour d'horizon des algorithmes de tris.

Opérations. — Les opérations intéressantes pour les tris :
— comparaisons ;
— affectation et échanges de variables.

Proposition 1.0.0.1 ($\mathbf{N}$ est totalement ordonné). —
Les propriétés suivantes indiquent que $\mathbf{N}$ est totalement ordonné avec $<$:

$$\begin{cases} \forall x \neq y,\, x < y \text{ ou } y < x, \\ \forall x, y, z,\, x < y \text{ et } y < z \implies < z. \end{cases}$$

Définition 1.0.0.5 (Trier). —
Trier un tableau ou une liste d'entiers : T de longueur $n \in \mathbf{N}$ c'est trouver une permutation $\sigma \in S_n$ telle que :

$$\forall i \leq n-1,\ T[\sigma(i)] \leq T[\sigma(i+1)].$$

Exemple. — Pour $T = (10, 9, 7, 6, 8)$ on a :

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 0 & 2 \end{pmatrix}.$$

> LEMME 1.0.0.1. —
> Pour tout $n \in \mathbf{N}^*$, toute permutation de S_n peut s'écrire comme un produit de transpositions.