Minorant de complexité pour les tris par comparaison

Manon Ruffini

Soit A un algorithme de tri par comparaison. Il résout le problème suivant :
Entrée : Un tableau $T = (a_1 \dots a_n)$
Sortie : $\bar{T} = (a_{\pi(1)} \dots a_{\pi(n)})$, où $\pi \in \mathfrak{S}_n$ telle que : $a_{\pi(1)} \leqslant \dots \leqslant a_{\pi(n)}$ [1]

Étape 1

Définition de l'arbre de décision.

— Un **arbre de décision** binaire est un arbre binaire qui représente les différentes exécutions de l'algorithme sur toutes les entrées d'une certaine taille
— Les **feuilles** de l'arbre sont les résultats des différentes exécutions : les permutations $(\pi(1), \dots, \pi(n))$ telles que $a_{\pi(1)} \leqslant \dots \leqslant a_{\pi(n)}$
— Les **nœuds** représentent les comparaisons entre les éléments effectués par l'algorithme. Pour un nœud "$a_i \leqslant a_j$", si $a_i \leqslant a_j$ l'exécution se poursuit dans le sous-arbre gauche ; si $a_i > a_j$ l'exécution se poursuit dans le sous-arbre droit.

A toute exécution de A correspond une branche de l'arbre. Le nombre de comparaisons correspond à la longueur de la branche.

Étape 2

L'arbre de décision associé à A sur une entrée de taille n a exactement $n!$ feuilles.

Démonstration : Il y a $|\mathfrak{S}_n| = n!$ ordres possibles pour les n éléments de l'entrée. Chaque feuille correspond à une exécution de l'algorithme ; et à chaque exécution correspond un ordre initial des éléments.
— Deux permutations ne peuvent apparaître dans une même feuille : il y a au moins $n!$ feuilles
— Toute permutation détermine un unique chemin dans l'arbre : il y au plus $n!$ feuilles ■

Étape 3

Borne inférieure pour la complexité en pire cas

On note $C(n)$ le nombre de comparaisons que l'on doit effectuer dans le pire cas, pour une entrée de taille n. La valeur $C(n)$ correspond à la longueur d'un chemin de la racine à une feuille de longueur maximale, i.e. la hauteur de l'arbre. Soit h la hauteur de l'arbre de décision. Alors :

$$n! \leqslant 2^h \text{ (récurrence sur } h\text{)}$$

Ainsi :

$$C(n) \geqslant \lg(n!)$$

Or, d'après la formule de Stirling : $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$, donc $\lg n! \sim n \lg n$. Finalement :

$$C(n) = \Omega(n \lg n)$$

Étape 4

Borne inférieure pour la complexité en moyenne

1. L'ordre relatif des éléments est donné par $\leqslant$

Les permutations de départs sont **équiprobables**. Donc, si on note $c(n)$ la complexité moyenne sur une entrée de taille n, on a :

$$c(n) = \text{ hauteur externe moyenne } = \frac{1}{\#\{\text{feuilles}\}} \sum_{f \text{ feuille}} (\text{hauteur de } f)$$

Soit B un arbre binaire ; Soit F son nombre de feuilles. Pour tout feuille f de B, on note $h_B(f)$ la hauteur de cette feuille dans B. De plus, on note $\mathrm{HE}(B)$ la hauteur externe moyenne de B. On a :

$$\mathrm{HE}(B) = \frac{1}{F} \sum_{f \text{ feuille}} h(f)$$

Montrons par induction sur B que $\mathrm{HE}(B) \geqslant \log_2(F)$:

- Si B est une feuille : $\mathrm{HE}(B) = 0 \geqslant \lg(1) = \lg(F)$
- Si $B = \begin{tikzcd} & \cdot \arrow[dl] \arrow[dr] & \\ B_1 & & B_2 \end{tikzcd}$, et si on note F_i le nombre de feuilles de l'arbre B_i, pour $i = 1, 2$, on a :

$$\begin{aligned}
\mathrm{HE}(B) &= \frac{1}{F} \sum_{f \text{ feuille}} h_B(f) \\
&= \frac{1}{F} \left(\sum_{f \text{ feuille de } B_1} (h_{B_1}(f) + 1) \sum_{f \text{ feuille de } B_2} (h_{B_2}(f) + 1) \right) \\
&= \frac{1}{F} \left(\underbrace{F_1 + F_2}_{=F} + F_1 \mathrm{HE}(B_1) + F_2 \mathrm{HE}(B_2) \right) \\
&= 1 + \frac{F_1}{F} \mathrm{HE}(B_1) + \frac{F_2}{F} \mathrm{HE}(B_2) \\
&\underset{HI}{\leqslant} 1 + \frac{F_1}{F} \lg(F_1) + \frac{F - F_1}{F} \lg(F - F_1)
\end{aligned}$$

On est donc amené à chercher le minimum de $f : x \mapsto 1 + \frac{x}{F} \lg x + \frac{F-x}{F} \lg(F - x)$. Or,

$$f'(x) = \frac{\lg x}{F} + \frac{cx}{Fx} - \frac{\lg(F - x)}{F} - \frac{C}{F} = \frac{1}{F} \lg \left(\frac{x}{F - x} \right)$$

Donc f' s'annule uniquement en $x = \frac{F}{2}$ et $f''(x) = \frac{C}{x(F-x)}$, donc $f''\left(\frac{F}{2}\right) = \frac{C}{(F/2)^2} > 0$. Donc f admet un unique minimum en $F/2$. Ainsi :

$$\mathrm{HE}(B) \geqslant f(F/2) = \lg(F)$$

On applique ce résultat à l'arbre de décision : $c(n) \geqslant \lg(n!) = \Theta(n \lg n)$. Finalement :

$$c(n) = \Omega(n \lg n)$$

Références

[1] Thomas H. Cormen, *Algorithmique*. Dunod, 3è édition, 2010.

[2] C. Froidevaux, M-C. Gaudel, M. Soria, *Types de données et algorithmes*. McGraw-Hill, 1990.