Calcul déterministe de la médiane

Benjamin Hellouin

Cormen, 9.3

On s'intéresse à la résolution du problème de sélection : trouver, dans un tableau, le i-ème plus petit élément (i étant donné en entrée). On montre que le paradigme diviser-pour-régner peut être utilisé pour donner un algorithme en temps linéaire, donc plus efficace que de trier le tableau puis prendre le i-ième élément. Une application majeure de ce résultat est la recherche d'un pivot pour l'algorithme de tri rapide : on peut par exemple rechercher la médiane basse du tableau, i.e. le $\lfloor \frac{n}{2} \rfloor$-ième élément, pour garantir un temps d'exécution en $O(n \log n)$ dans le pire cas (et non pas en moyenne).

L'algorithme procède comme suit :
- Diviser le tableau d'entrée en $\lfloor \frac{n}{5} \rfloor$ tableaux de 5 éléments, avec éventuellement un dernier tableau de ≤ 5 éléments ;
- Trier chaque tableau et en extraire la médiane (en temps constant) ;
- Appliquer récursivement cet algorithme pour trouver la médiane de ces médianes ;
- Partitionner le tableau autour de cette dernière médiane, obtenant un tableau gauche de taille $k-1$;
  - si $k = i$, renvoyer la médiane trouvée ;
  - si $i < k$, appliquer l'algorithme sur le tableau de gauche et i ;
  - si $k < i$, appliquer l'algorithme sur le tableau de droite et $k - i$.

Pour faire l'analyse de complexité, on regarde le nombre d'éléments plus grands que l'élément renvoyé x : pour au moins la moitié des tableaux, 3 éléments au moins sont plus grands que x, en excluant celui contenant x (seulement 2) et le tableau excédentaire, soit

$$3\left(\left\lceil \frac{1}{2}\left\lfloor \frac{n}{5} \right\rfloor \right\rceil - 1\right) + 2 \geq \frac{3n}{10} - 3.$$

et symétriquement, de sorte que l'algorithme est rappelé récursivement sur un tableau de taille au plus $\frac{7n}{10} + 3$. Ceci nous permet d'étudier la complexité de l'algorithme.

$$\begin{aligned} T(n) \leq \quad & O(n) && \text{(étapes 1,2,4)} \\ & + T(\lfloor n/5 \rfloor) && \text{(étape 3)} \\ & + T(\tfrac{7n}{10} + 3) && \text{(étape 4 bis, ter)} \end{aligned}$$

Le théorème de Master nous indique que cette récurrence implique un temps d'exécution linéaire. On peut s'en convaincre en supposant que le terme non récursif est majoré par an, et en prouvant par récurrence forte que $T(n) < cn$ pour un certain c. La relation de récurrence devient, par hypothèse de récurrence :

$$T(n) \leq an + cn/5 + c(7n/10 + 3) = (a + c/5 + 7c/10)n + 3c$$

soit $T(n) \leq cn$ si $an - cn/10 + 3c < 0$. Si on prend $c > 20a$, ceci devient $-cn/20 + 3c < 0$, ce qui est vrai pour $n > 60$. en prenant c assez grand pour que $T(n) \leq cn$ pour $n \leq 60$, la récurrence est établie et on prouve que l'algorithme de sélection est en temps linéaire.

Application. On peut choisir le pivot de l'algorithme de tri rapide en utilisant l'algorithme de sélection de la médiane décrit ci-dessus, de sorte que $C(n) = T(n) + 2C(n/2)$, ce qui suffit à obtenir un temps $O(n \log n)$ dans le pire cas, ce qui est la meilleure borne asymptotique atteignable. Dans la pratique, rechercher la médiane à chaque étape est trop laborieux. On préfèrera utiliser la structure précédente et choisir pour pivot la médiane des médianes, ce qui conserve la borne asymptoptotique en $O(n \log n)$ mais réduit le temps de calcul non récursif (en réduisant la qualité du pivot).