Médiane en temps linéaire

Léo Gayral

2017-2018

ref : Cormen – Introduction to algorithms, third edition – p.220

Remarque 1. En pratique, lorsqu'on utilise un algorithme de tri, on n'utilise pas toute l'information donnée par le résultat. S'il est plus naturel de considérer les quantiles au sein d'une liste déjà triée, il peut cependant être plus efficace d'extraire ces quantiles d'une liste quelconque de façon astucieuse plutôt que de la trier entièrement.

Ce principe est particulièrement apparent si on s'intéresse au minimum d'une liste, auquel cas il suffit de parcourir la liste une seule fois, en gardant une trace du minimum sur les valeurs déjà visitées.

Cette idée se généralise mal, car pour obtenir la valeur du $k^{ième}$ élément, on a besoin de stocker la sous-liste triée des k plus petits éléments, ce qui induit moralement une complexité en $O(n\ln(k))$. C'est typiquement le cas de la médiane, de l'élément d'indice $k \sim \frac{n}{2}$.

On va par la suite considérer un algorithme proche de $QUICKSORT$, avec un pivot, mais où on ne va pas chercher à trier la partie de la liste qui ne nous intéresse pas.

Proposition 1. On définit l'algorithme $SELECTION$, qui prends en entrée une liste injective L et $i \in [\![1, |L|]\!]$ un indice, et retourne le $i^{ième}$ élément de L en sortie, comme suit :

```
1  def SELECTION(L, i) :
2          q := PIVOT(L)
3          G := {x ∈ L, x < q}
4          D := {x ∈ L, x > q}
5          si i = |G| + 1 :
6              retourner q
7          sinon si i ≤ |G| :
8              retourner SELECTION(G, i)
9          sinon :
10             retourner SELECTION(D, i − 1 − |G|)
```

où $PIVOT$ prends une liste L en entrée et renvoie un certain $q \in L$ en sortie.

Indépendamment du pivot choisi, $SELECTION$ termine et renvoie bien l'élément voulu.

Démonstration.
Le résultat se montre par récurrence sur $|L|$. En effet, si $|L| = i = 1$, alors q est l'élément voulu, et $G = D = \emptyset$ donc $|G| + 1 = 1 = i$. D'autre part, comme $\max(|G|, |D|) \leq |L| - 1$, on peut appliquer l'hypothèse de récurrence aux appels récursifs, d'où le résultat. □

Théorème 1. Pour un choix de $PIVOT$ adapté, $SELECTION$ a une complexité linéaire dans le pire des cas.

Démonstration.
Idéalement, on aimerait un choix de pivot équilibré, proche de la médiane. On va pour cela regrouper les éléments par paquets de taille 5, avec éventuellement un paquet de taille $n - 5\left\lfloor \frac{n}{5} \right\rfloor$.

On considère $MEDIANE(L)$, qui trie L puis renvoie l'élément en position $\left\lfloor \frac{n}{2} \right\rfloor$, en temps *constant* sous la condition $|L| \leq 5$. On peut alors définir l'algorithme $PIVOT$ par :

```
1  def PIVOT(L) :
2      si |L| ≤ 5 :
3          retourner MEDIANE(L)
4      sinon :
5          M := ∅
6          pour i de 0 a ⌈|L|/5⌉ − 1 :
7              ajouter MEDIANE(L[(5i + 1) : min(5i + 5, |L|)]) a M
8          retourner SELECTION(M, ⌊|M|/2⌋)
```

On est certain de ne pas avoir une infinité d'appels imbriqués entre $PIVOT$ et $SELECTION$, par décroissance de la taille des entrées, donc $PIVOT$ termine bien sur toute entrée. On va maintenant estimer la position effective du pivot obtenu dans L, ce qui nous permettra d'évaluer la complexité de $SELECTION$.

Pour chaque paquet de 5 éléments, trois sont supérieurs ou égaux à leur médiane. On a $\left\lceil \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rceil$ paquets dont la médiane est supérieure ou égale à celle de M. Quitte à oublier le paquet au centre de M (dont seulement deux éléments sont plus grands strictement que le pivot) et celui à droite (qui n'a pas forcément 5 éléments), le nombre d'éléments de L strictement plus grands que le pivot, la médiane de M, vérifie :

$$|D| \geq 3 \times \left(\left\lceil \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rceil - 2 \right) \geq \frac{3n}{10} - 6\,.$$

On obtient la même minoration sur $|G|$ avec un raisonnement similaire, donc en passant au complémentaire, on a la majoration :

$$\max(|G|, |D|) \leq \frac{7n}{10} + 6\,.$$

On considère alors T la complexité en comparaisons de $SELECTION$ dans le pire des cas ($T(n) = \max_{|L|=n} \max_{1 \leq i \leq n} [\dots]$), et C celle de $PIVOT$. Sous l'hypothèse raisonnable que T est croissante, avec notre définition jointe de $SELECTION$ et $PIVOT$, on a :

$$\begin{array}{rcl} C(n) & \leq & \left\lceil \frac{n}{5} \right\rceil \times O(1) + T\left(\left\lceil \frac{n}{5} \right\rceil\right) \\ T(n) & \leq & C(n) + T\left(\frac{7n}{10} + 6\right) \end{array}$$

d'où $T(n) \leq T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\frac{7n}{10} + 6\right) + \alpha n$.

Montrons par récurrence que $T(n) \leq \beta n$. Si le résultat est vrai jusqu'au rang $n-1$ alors :

$$\begin{array}{rcl} T(n) & \leq & \beta \left\lceil \frac{n}{5} \right\rceil + \beta \left(\frac{7n}{10} + 6\right) + \alpha n \\ & \leq & \frac{9\beta}{10} n + \alpha n + 7\beta \\ & \leq & \beta n + \left(7\beta + \left[\alpha - \frac{\beta}{10}\right] n\right) \end{array}$$

donc, dès lors que $n \geq 140$, il suffit de considérer $\beta \geq 20\alpha$ pour avoir

$$\beta \times \left(\frac{n}{10} - 7\right) \geq \frac{\beta n}{20} \geq \alpha n$$

et donc l'hérédité. Quitte à prendre une constante compatible avec les rangs $n < 140$, on peut initialiser la récurrence, d'où $T(n) = O(n)$. □

Remarque 2. L'algorithme précédent peut être réutilisé comme choix d'un pivot au sein d'un tri rapide. Ce faisant, on peut garantir une complexité asymptotique en $O(n \ln(n))$ en moyenne et dans le pire des cas.

Cependant, en pratique, le choix d'un pivot aléatoire est plus efficace car il évite le *calcul* du pivot précédent.