Esiee-Paris - unité d'algorithmique IGI2102 - feuille numéro 10
p-ème valeur

Décembre 2023– R. Natowicz, I. Alame, A. Çela, X. Hilaire, T. Wu, W. Xu

Le calcul de la p-ème valeur d'une suite de n valeurs est une question de base dans nombre de domaines, en particulier en statistiques. Par exemple, la médiane est le cas $p = \frac{n}{2}$.

Une méthode évidente pour calculer la p-ème valeur est de trier les n valeurs , ce calcul est en $\Theta(n \log n)$, puis de retourner la p-ème valeur, ce calcul est en $\Theta(1)$. La complexité de cette méthode est donc $\Theta(n \log n)$.

À la suite de son algorithme QuickSort de tri rapide d'un tableau, C.A.R. Hoare a proposé l'algorithme QuickSelect pour calculer la p-ème valeur en $\Theta(n)$. Tout comme QuickSort, QuickSelect est un algorithme "randomisé".

Dans ce qui suit, nous cherchons à calculer la p-ème valeur du tableau d'entiers $T[0 : n]$, avec $p \in [0 : n]$. Exemples : la 0-ème valeur est la valeur minimum du tableau, la $n-1$-ème valeur est sa valeur maximum.

Remarque : les Sapiens demanderont plutôt le calcul de la s-ème valeur, $s \in [1 : n+1]$, car pour un Sapiens la 1-ère valeur est la valeur minimum et la n-ème est la valeur maximum. Les Sapiens sont bizarre. Nous n'avons pas le choix. *Il faut faire avec.* "Du coup", quand une ou un Sapiens demandera le calcul de la s-ème valeur, nous appellerons notre fonction avec $s-1$, nous lui enverrons cette valeur et tout le monde sera content : l'algorithme, l'ordinateur, les Sapiens, et nous.

Nous connaissons la fonction `int segmenter(int[] T, int i, int j)` du QuickSort "randomisé". L'appel `int k = segmenter(T,i,j)` calcule une permutation des valeurs du sous-tableau $T[i : j]$ vérifiant $T[i : k] \leq T[k] < T[k+1 : j]$ et retourne et l'indice k. Cette fonction commence par choisir au hasard un indice h dans l'intervalle $[i : j]$, elle permute les valeurs t_i et t_h, puis elle calcule la permutation des valeurs de $T[i : j]$ et retourne l'indice k.

Exercice 17. Calcul de la p-ème valeur.
Nous voulons la p-ème valeur du sous-tableau $T[i : j]$, $p \in [0 : j-i]$.

– Si $j - i = 1$, nous connaissons la p-ème valeur de $T[i : j]$. En effet : $p \in [0 : 1]$, donc $p = 0$, donc t_i est la $p-$ème valeur.

– Si $j - i > 1$, que faire ? Nous pouvons segmenter le sous-tableau $T[i : j]$. Bonne idée. Alors deux situations peuvent se présenter :

1. $p + i < k$. Quelle est la p-ème valeur de $T[i : j]$? (Donner son expression en fonction d'un sous-problème de taille inférieure à $j - i$.)
2. $p + i \geq k$. Même question.
3. Écrire la fonction `qsel(int[] T, int i, int j, int p)` qui retourne la p-ème valeur de $T[i : j]$, $0 \leq p < j - i$.

On démontre que la complexité de `quickSelect` est $\Theta(n)$ en résolvant l'équation de récurrence du temps de calcul de l'appel `qsel(T, 0, n, p)` :

- $t(1) = \mathrm{c}$, temps constant
- $t(n > 1) = (a \times n + b) + t(n/2)$

  où $a \times n + b$ est le temps de la segmentation du sous-tableau $T[i : j]$ de taille $n = j - i$.

Nous ferons la démonstration en cours (en attendant : le faire en exercice personnel.)