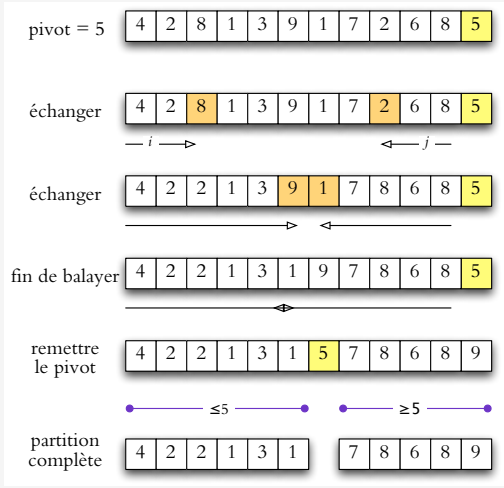
14 Tri rapide 2



Algo QUICKSORT$(A[0..n-1], g, d)$ *// tri de $A[g..d]$*
Q1 **if** $g \geq d$ **then return** *// cas de base*
Q2 $i \leftarrow$ PARTITION(A, g, d)
Q3 QUICKSORT$(A, g, i-1)$
Q4 QUICKSORT$(A, i+1, d)$

Algo PARTITION(A, g, d) *// partition de $A[g..d]$*
P1 chosir le pivot $p \leftarrow A[d]$
P2 $i \leftarrow g-1$; $j \leftarrow d$
P3 **loop**
P4 **do** $i \leftarrow i+1$ **while** $A[i] < p$
P5 **do** $j \leftarrow j-1$ **while** $j \geq i$ et $A[j] > p$
P6 **if** $i \geq j$ **then** sortir de la boucle
P7 échanger $A[i] \leftrightarrow A[j]$
P8 échanger $A[i] \leftrightarrow A[d]$
P9 **return** i

Pour trier un tableau $A[0..n-1]$ en ordre croissant, on exécute QUICKSORT$(A, 0, n-1)$.

14.1 Améliorations

Petits sous-tableaux. Le **tri par insertion** est plus rapide que quicksort quand $d-g$ est petit ($g \geq d-\ell^*$ avec $\ell^* = 5..20$). En Ligne Q1, c'est mieux donc de faire le tri par insertion pour tels petits tableaux. En fait, on peut juste **ignorer** les petits sous-tableaux entièrement (retourner si $g \geq d-\ell^*$ en Ligne Q1). À la fin, il faut parcourir le tableau entier selon tri par insertion en $\Theta(n\ell^*) = \Theta(n)$.

Choix du pivot. Deux choix performent très bien en pratique : médiane ou aléatoire.

Médiane de trois

P1.1 **si** $d \geq g+2$ **alors**
P1.2 **if** $A[g] > A[d-1]$ **then** échanger $A[g] \leftrightarrow A[d-1]$
P1.3 **if** $A[d] > A[d-1]$ **then** échanger $A[d] \leftrightarrow A[d-1]$
P1.4 **if** $A[g] > A[d]$ **then** échanger $A[g] \leftrightarrow A[d]$
P1.5 $p \leftarrow A[d]$ *// $A[g] \leq A[d] \leq A[d-1]$*

et on se sert des **sentinelles** qui sont maintenant en place $A[g]$, $A[d-1]$:

P2' $i \leftarrow g; j \leftarrow d-1$
P5' **do** $j \leftarrow j-1$ **while** $A[j] > p$

Aléatoire

P1.1 $k \leftarrow$ RANDOM(g, d)
P1.2 $p \leftarrow A[k]$
P1.3 **if** $k \neq d$ **then**
P1.4 $A[k] \leftarrow A[d]$
P1.5 $A[d] \leftarrow p$

14.2 Moyen cas

Théorème 14.1. *Soit $D(n)$ le nombre moyen de comparaisons avec un pivot aléatoire, où n est le nombre d'éléments dans un tableau $A[0..n-1]$. Alors,*

$$\frac{D(n)}{n} = O(\log n).$$

Lemme 14.2. *On a $D(0) = D(1) = 0$, et*

$$D(n) = n - 1 + \frac{1}{n}\sum_{i=0}^{n-1}\Big(D(i) + D(n-1-i)\Big) = n - 1 + \frac{2}{n}\sum_{i=0}^{n-1} D(i).$$

Démonstration. Supposons que le pivot est le i-ème plus grand élément de A. Le pivot est comparé à $(n-1)$ autre éléments pour la partition. Les deux partitions sont de tailles i et $(n-1-i)$. Or, i prend les valeurs $0, 1, \ldots, n-1$ avec la même probabilité. ■

Preuve de Théorème 14.1. Par Lemme 14.2,

$$\begin{aligned} nD(n) - (n-1)D(n-1) &= \Big(n(n-1) + 2\sum_{i=0}^{n-1} D(i)\Big) - \Big((n-1)(n-2) + 2\sum_{i=0}^{n-2} D(i)\Big) \\ &= 2(n-1) + 2D(n-1). \end{aligned}$$

D'où on a

$$\frac{D(n)}{n+1} = \frac{D(n-1)}{n} + \frac{2n-2}{n(n+1)} = \frac{D(n-1)}{n} + \frac{4}{n+1} - \frac{2}{n}.$$

Avec $E(n) = \frac{D(n)-2}{n+1}$, on a $E(n) = E(n-1) + \frac{2}{n+1}$, donc $E(n) = E(0) + \frac{2}{2} + \frac{2}{3} + \cdots + \frac{2}{n+1} = 2H_{n+1} - 4$, où $H_n = \sum_{i=1}^{n} 1/i = \ln n + \gamma + o(1)$ est le n-ème nombre harmonique ($\gamma = 0.5772\cdots$ est la constante d'Euler-Mascheroni).

En retournant à $D(n) = 2 + (n+1)E(n)$, on a alors

$$D(n) = 2(n+1)H_{n+1} - 4n - 2 < 2nH_{n+1}$$

Donc le nombre de comparaisons en moyenne est tel que $\frac{D(n)}{n} < 2H_{n+1} = O(\log n)$. ■

En fait la preuve montre que $D(n)/n = \big(2 + o(1)\big)H_n \sim 2\ln n \approx 1.39 \lg n$. C'est seulement 39% pire que le tri le plus rapide imaginable (qui fait $\lg n!$ comparaisons au pire).