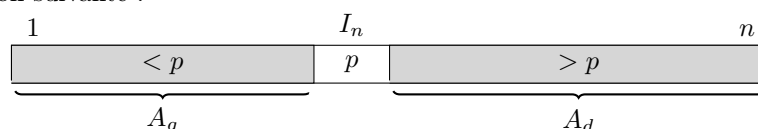


Algorithme 1 : Tri rapide.**Entrée** : $A[\ell..r]$ un tableau d'éléments comparables.**Partition**(A, ℓ, r)**Données** : p un élément de A , i, j des indices.**Sortie** : $i \in [\ell, r]$, A étant réarrangé pour que $A[\ell..k[\leq p = A[k] \leq A]k..r]$. $p \leftarrow A[r]$; $i \leftarrow \ell$ **pour** $j = \ell$ à $r - 1$ **faire** **si** $A[j] \leq p$ **alors** échanger $A[i] \leftrightarrow A[j]$ $i \leftarrow i + 1$ échanger $A[i] \leftrightarrow A[r]$ **retourner** i **TriRapide**(A, ℓ, r)**si** $\ell < r$ **alors** $i \leftarrow \text{Partition}(A, \ell, r)$ **TriRapide**($A, \ell, i - 1$); **TriRapide**($A, i + 1, r$)

Théorème 1. Soit C_n le nombre moyen de comparaisons effectuées par l'algorithme de tri rapide (voir [algorithme 1](#)) sur un tableau uniformément distribué de taille n . Alors $C_n = \Theta(n \log_2 n)$.

Preuve. Lors de la première étape de partitionnement, l'algorithme effectue $n - 1$ comparaisons pour réorganiser les éléments de la façon suivante :



Le pivot p et donc sa position finale I_n suivent une loi uniforme. Aussi, sachant $I_n = i \in \{1, \dots, n\}$, les sous-tableaux A_g et A_d sont uniformément distribués. En effet chaque couple (A_g, A_d) possible peut provenir de $\binom{n-1}{i-1}$ permutations initiales du tableau entier donc a une probabilité $1/((i-1)!(n-i)!)$, ce qui correspond bien à la probabilité uniforme sur $\mathfrak{S}_{i-1} \times \mathfrak{S}_{n-i}$. Ainsi $C_0 = 0$ et pour tout $n \geq 1$,

$$C_n = \sum_{i=1}^n \frac{1}{n} (C_{i-1} + C_{n-i}) + n - 1,$$

donc

$$\begin{aligned} nC_n &= \sum_{i=1}^{n-1} 2C_i + n(n-1) \\ &= 2C_{n-1} + \left[\sum_{i=1}^{n-2} 2C_i + (n-2)(n-1) \right] + 2(n-1) \\ &= (n+1)C_{n-1} + 2(n-1), \end{aligned}$$

puis

$$\frac{C_n}{n+1} = \frac{C_{n-1}}{n} + \frac{2(n-1)}{n(n+1)} = \sum_{k=1}^{n-1} \frac{2(k-1)}{k(k+1)} = 2 \log(n) + O(1).$$

D'où $C_n = 2n \log(n) + O(n)$. ■**Références.** [CLRS]**902** Diviser pour régner : exemples et applications.**903** Exemples d'algorithmes de tri. Complexité.**926** Analyse des algorithmes : complexité. Exemples.

[CLRS] Thomas H. CORMEN, Charles E. LEISERSON, Ronald L. RIVEST et Clifford STEIN : *Algorithmique*. Dunod, 3^{ème} édition.