Complexité moyenne du tri rapide

Leçons 903, 926, 931

Références Éléments d'algorithmique, Beauquier p. 126.

Théorème

La complexité en moyenne du tri rapide sur la distribution uniforme des permutations de L est $O(n \log n)$

Résumé

I - Description de l'algorithme de tri rapide

II - La procédure de pivotage se fait en au plus n+1 comparaisons.

III - Après pivotage, les permutations à gauche (/à droite) sont équiprobables.

IV - Analyse de la complexité moyenne

I • L'algorithme classe récursivement et en place les éléments en deux listes : les éléments plus petits et ceux plus grands qu'un pivot.

Pour un tableau T[i..j], on définit récursivement l'algorithme de tri rapide :

```
TRI_RAPIDE(T[i...j])
    Si i < j
        k = PIVOTER(T[i..j])
        TRI_RAPIDE(T[i..(k-1)])
        TRI_RAPIDE(T[(k+1)..j])
```

• La procédure PIVOTER est là où le tri s'effectue, et il repose sur une propriété des couples inversés par rapport au pivot γ, c'est-à-dire deux indices s et t tels que $i \leq s < t \leq j$

et $T[s] > \gamma \geq T[t]$

On peut donc trouver un couple inversé dans $T[i,j]$ avec :

```
COUPLE_INVERSÉ(T[i..j], γ)
| s = i , t = j
| Tant que s ≤ j et T(s) ≤ γ  :  s = s + 1
| Tant que t ≥ i et T(t) > γ  :  t = t - 1
| Renvoyer (s,t)
```

Un couple renvoyé par cet algorithme est inversé, et on est sûr qu'il n'y en a eu aucun avant (pour $s' < s$ et $t' > t$)
(mais pas $s' = s$ et $t' = t$)

• La procédure de PIVOT se définit alors ainsi sur $T[i..j]$

- prendre $\gamma := T[i]$ comme pivot
- chercher un premier couple inversé (s,t) avec PREMIER_COUPLE
- inverser les deux et recommencer sur $T[s+1..t-1]$, jusqu'à ce qu'il soit réduit à un élément ou à l'ensemble vide.
- mettre le pivot à sa place en le mettant en position $t-1$.

$^{\gamma}$6	$\boxed{14}$	3	1	8	4	$\boxed{5}$	9
$^{\gamma}$6	5	3	1	$\boxed{8}$	$\boxed{4}$	14	9
4	5	3	1	$^{\gamma}$6	8	14	9

II • Le PIVOT se fait en moins de $n+1$ comparaisons :

$t-s$ vaut au plus n, et est décrémenté chaque fois qu'une comparaison est faite. Comme on s'arrête quand il est ≤ 0, on a au plus $n+1$ comparaisons.

• Supposons que la permutation initiale de T est choisie de façon uniforme des listes

III • Montrons qu'après pivotage, les distributions à gauche et à droite du pivot sont uniformes

• Soit r le rang du pivot, on associe des permutations de $[\![1..r-1]\!]$ et $[\![r+1..n]\!]$ aux listes gauche et droite, notées π_1 et π_2.

• supposons que $1 < r < n$, car sinon PIVOT n'a juste rien changé.

• On pose $\pi = \pi_1 \cdot (r) \cdot \pi_2$, et on va regarder combien de permutations initiales de $[\![1..n]\!]$ mènent à π après PIVOT.

→ supposons que l'algorithme a échangé q couples

$(s_1, t_1), (s_2, t_2), \dots, (s_q, t_q)$

où $1 \leq s_1 < s_2 < \dots < s_q \leq r-1$ et $r+1 \leq t_1 < t_2 < \dots < t_q \leq n$. (*)

→ posons τ_0 la transposition qui échange $\pi(1)$ et $\pi(r)$

et τ_k la transposition qui échange $\pi(s_k)$ et $\pi(t_k)$

alors $\tau_q \circ \tau_{q-1} \circ \dots \circ \tau_0 \circ \pi$ déroule l'algorithme à l'envers et redonne la permutation initiale du tableau.

→ on peut donc appliquer ça à π : on choisit $s_1..s_q$ et $t_1..t_q$ vérifiant (*), alors l'algorithme de pivot sur $\pi' = \tau_q \circ \dots \circ \tau_0 \circ \pi$ redonnera π.

+ la fonction $s_1, \dots, s_q, t_1, \dots, t_q \to \pi'$ est injective

⇒ le nombre de π' qui convient est donc déterminé par : le choix de q puis des s_k et t_k, il y en a donc :

$\left\{\begin{array}{l} q \leq r-1 \\ \text{et } q \leq n-r \end{array}\right.$ ⟶

$$\sum_{q=1}^{\min(r-1,n-r)} \binom{r-1}{q}\binom{n-r}{q}$$

choix des s_k choix des t_k

Cela ne dépend pas de la permutation π, on a donc autant de possibilités pour chacune des permutations π (donc pour π_1 et π_2 aussi !)

IV] • L'hypothèse d'équiprobabilité est conservée donc, en posant $M(n)$ le nombre de comparaisons pour une liste de taille n, on a :

$$M(n) \leqslant \sum_{r=1}^{n} \frac{1}{n}\Big[\underbrace{n+1}_{\text{PIVOT}} + \underbrace{M(r-1) + M(n-r)}_{\text{tris récursifs}} \Big]$$

Or $\displaystyle\sum_{r=1}^{n} M(r-1) = \sum_{r=1}^{n} M(n-r) = \sum_{r=1}^{n-1} M(r)$

D'où $\displaystyle M(n) \leqslant n+1 + \frac{2}{n} \sum_{r=1}^{n} M(r)$

et $\displaystyle M(n) = O\Big(n \sum_{k=1}^{n+1} \frac{1}{k}\Big)$

$= O(n \log n)$.