

Algorithmique avancée

Série d'exercices n° 3

Exercice 1 :

Dans cet exercice, on va supposer que le coût en temps du tri d'un tableau par insertion est $c.n^2$ pour un tableau de taille n (c étant une constante). On effectue l'algorithme hybride suivant :

- Diviser un tableau non trié de taille n en k sous-tableaux de taille égale (n/k).
- Trier chaque sous-tableau en utilisant le tri par insertion.
- Fusionner les sous-tableaux obtenus.

1. Calculer le temps de la fusion en fonction de k et de n .
2. En déduire le temps que mettra l'algorithme hybride donné ci-dessus.
3. Conclure en comparant avec le tri fusion et le tri par insertion. (Veuillez tenir compte que k peut être une fonction de n .)

Exercice 2 :

Dans cet exercice, on note $T(n)$ le temps nécessaire pour un algorithme de tri pour trier un tableau de taille n . On suppose que T vérifie :

$$T(1) = 0 \text{ et } T(n) = 3T(n/3) + 2cn$$

où c est une constante positive.

1. Montrer que $T(3^m) = 2c.m.3^m$ ($m \neq 0$).
2. En déduire que $T(n) = \mathcal{O}(n \log n)$.

Exercice 3 :

On a un tableau T de n entiers dans $\{0, 1, 2\}$.

1. Donner un algorithme linéaire pour trier T .
2. Généraliser au cas des entiers dans $\{0, 1, \dots, k\}$. Discutez explicitement des cas $k = \mathcal{O}(\log n)$ et $k = \Omega(\log n)$.

Exercice 4 :

On a deux tableaux X et Y contenant chacun n entiers. Etant donné une valeur entière α , écrire un algorithme pour décider (répondre oui ou non) si α est la somme d'un élément de X et d'un élément de Y .