Feuille de TD n°3 : diviser pour régner

MPSI option informatique

Mars 2025

Exercice 1 : Ecrire en Caml une fonction qui calcule x^n pour x et n entiers, utilisant l'exponentiation rapide.

On proposera une version itérative et une version récursive.

Exercice 2 : Programmer une fonction de fusion de deux segments adjacents triés d'un même tableau, en un segment de tableau trié *(on pourra utiliser un tableau auxiliaire)*.

En déduire un programme mettant en œuvre l'algorithme de "tri fusion" d'un tableau.

Exercice 3 : *Tri par fusion d'une liste* :

1) Écrire une procédure de partage d'une liste en deux listes de tailles égales (éventuellement à 1 près !), puis une fonction réalisant la fusion de deux listes triées.

2) En déduire un programme de tri par fusion d'une liste. En évaluer la complexité.

Exercice 4 : Écrire un programme déterminant le maximum d'un tableau par une méthode « diviser pour régner ». Évaluer sa complexité.

Exercice 5 : Calcul des termes de la suite de Fibonacci.

La suite de Fibonacci est définie par les relations : $F_0 = F_1 = 1$ et pour tout $n \geq 0$, $F_{n+2} = F_{n+1} + F_n$.

1) Ecrire un programme récursif calculant F_n pour $n \geq 0$.

Déterminer le nombre d'appels à la fonction selon n. On posera T(n) le nombre d'appels à la fonction pour le calcul de F_n, puis on donnera une relation entre $T'(n) = T(n) + 1$, $T'(n-1)$ et $T'(n-2)$ pour $n \geq 2$.

2) Ecrire un programme récursif calculant le couple (F_{n-1}, F_n).

Déterminer le nombre d'appels à la procédure selon n.

3) Démontrer la relation : $\forall (n,p) \in \mathbb{N}^{*2}$, $F_{n+p} = F_n F_p + F_{n-1} F_{p-1}$.

En déduire un programme de calcul de F_n selon la méthode "diviser pour régner".

Déterminer le nombre d'appels à la procédure selon n.

Exercice 6 : Une médiane d'un ensemble E de cardinal n est un élément $m \in E$ tel que

$$Card\,\{x \in E / x < m\} = \left\lfloor \frac{n}{2} \right\rfloor$$

1) Montrer l'unicité de la médiane d'un ensemble.

2) Donner un algorithme de complexité $\Theta(n^2)$ (mesurée en comparaisons) pour trouver la médiane d'un tableau.

3) Donner un algorithme de complexité $\Theta(n)$ (mesurée en comparaisons) pour trouver la médiane de deux tableaux ordonnés de même longueur n.

Exercice 7 : Programmer le tri rapide sur un tableau. La fonction programmée doit modifier le tableau en entrée, sans jamais créer de nouveau tableau. La partition doit effectuer un nombre d'opération linéaire en la taille du tableau.

Exercice 8 : Étant donnée une suite finie d'entiers $x = (x_1, \ldots, x_n)$, on appelle inversion de x tout couple (i, j) tel que $i < j$ et $x_i > x_j$. Par exemple, $(2, 3, 1, 5, 4)$ possède 3 inversions : les couples $(1, 3)$, $(2, 3)$, $(4, 5)$. On s'intéresse au calcul du nombre d'inversions de x.

1) Rédiger en Caml l'algorithme naïf, et montrer que son coût est un $\Theta(n^2)$. On représentera les suites finies d'entiers par le type `int array`.

2) Adopter une méthode « diviser pour régner »pour faire mieux (indication : adapter l'algorithme de tri fusion).