Informatique Petite Classe 1

9 novembre 2000

Exercices et programmation

Exercice 1.— Donner un algorithme qui trouve le plus petit et le plus grand élément d'un tableau de n nombres en effectuant un nombre total de comparaisons de l'ordre de $3n/2$.

Exercice 2.— (*Maximum des sommes partielles*, HC96) Etant donné une suite $(a_0, \ldots, a_{n-1})$ d'entiers, trouver le maximum des sommes

$$s_{i,j} = \sum_{k=i}^{j-1} a_k \qquad (0 \leq i \leq j \leq n)$$

Par exemple, pour $(-2, 11, -4, 13, -5, 2)$, le résultat est 20. Ecrire une fonction `int MSP(int [] a)` efficace pour évaluer cette quantité.

Exercice 3.— Ecrire une classe `Chrono` qui contient trois fonctions `start`, `stop` et `getElapsedTime`, et qui permet de mesurer le temps écoulé entre le dernier start, et le dernier stop. On utilise la fonction `long System.currentTimeMillis()`.

Dans les exercices qui suivent, une *permutation* est donnée dans un tableau $a[0 \ldots n-1]$ d'entiers compris entre 0 et $n-1$ et deux-à-deux distincts. Une *inversion* dans une permutation est un couple (i, j) tel que $i < j$ et $a[i] > a[j]$.

Exercice 4.— L'*inverse* d'une permutation a est la permutation b telle que

$$a[i] = j \Longleftrightarrow b[j] = i \ .$$

(a). Donner une fonction qui donne comme résultat l'inverse d'une permutation a.
(b). Ccomparer le nombre d'inversions de a et de son inverse.

Exercice 5.— Donner un algorithme qui engendre une permutation au hasard, de manière que chacune des $n!$ permutations soit équiprobable.

La *table d'inversions* d'une permutation a est le tableau t donné par

$$t[i] = \text{Card}\{j \mid (i, j) \text{ est une inversion}\} \ .$$

Par exemple, pour la permutation $a = 2, 3, 7, 5, 1, 9, 8, 4, 6$ on trouve $t = 1, 1, 4, 2, 0, 3, 2, 0, 0$.

Exercice 6.— Ecrire une fonction qui calcule le tableau t à partir du tableau a.

Exercice 7.— On demande d'écrire une fonction qui calcule a à partir de sa table d'inversions t. Pour cela, on calcule successivement l'entier figurant en position 1, puis 2, ... puis n dans le tableau a, en appliquant l'algorithme suivant:

- On construit une suite auxiliaire $s = 1, 2, 3, \ldots n$.
- On place $t[1] + 1$ en position 1 (`(a[1]:= t[1] + 1)` et on supprime cet élément de la suite s.
- On place le $t[2] + 1^{eme}$ élément de la suite s en position 2 et on le supprime de cette suite.
- Pour i de 3 jusqu à $n-1$, on place un élément (à déterminer) en position i et on le supprime de la suite s.
- On place le dernier élément qui reste dans la suite s en position n.