

TP - Les tableaux

3 novembre 2009

1 Parcours de tableau

1.1 Parcours simple

1.1.1 Parcours complet(1)

Soit un tableau d'entiers de taille n contenant n valeurs, écrire les programmes suivants :

- saisir les éléments du tableau
- afficher tous les éléments contenus dans le tableau
- calculer la somme des éléments du tableau
- calculer la somme des indices
- recherche du max

1.1.2 Parcours complet (2)

Soit un tableau d'entiers $[1..n]$ contenant NB valeurs ($NB < ou = à n$) :

- compter le nombre d'occurrences pour un entier saisi par l'utilisateur

1.1.3 Parcours complet (3)

Soit un tableau 'tab' d'entiers positifs contenant NB valeurs ($NB < ou = à n$) :

- créer un tableau 'toccure' tel que $toccure[i]$ vaut le nombre d'occurrences de 'i' dans 'tab'.

1.1.4 Parcours partiel

On souhaite remplacer dans un tableau $tabDest$ les $nbVal$ caractères enregistrés à partir de l'indice $indD$ par les $nbVal$ caractères d'un tableau $tabSource$ enregistrés à partir de l'indice $indS$. (voir exemple) Ecrire un algorithme qui effectue ce traitement (parcours de deux sous-tableaux)

Exemple : $nbVal : 3$

$tabDest : ABCDEFGHIJ$ $indD = 4$

$tabSource : MNOPQRST$ $indS = 5$

Après l'appel de votre algo, `tabDest` sera *ABCQRSGHJJ*, la sous-chaîne *DEF* ayant été remplacée par *QRS*.

1.1.5 Inversion, symétrie, doublons

Dans un tableau d'entiers $[0..n-1]$ contenant NB entiers ($NB < \text{ou} = n$), écrire les algorithmes suivants :

- inverser le tableau
 - en passant par un tableau intermédiaire ;
 - sans tableau intermédiaire.
- tester si le tableau est SYMETRIQUE : tableau lu de la même façon de gauche à droite et de droite à gauche. Exemple : 6 1 8 -1 6. (cas particulier du palindrome dans lequel les blancs ne sont pas pris en compte) ;
- compter le nombre d'éléments présents au moins deux fois dans le tableau.

1.2 Crible d'eratostene

Le but de cet algorithme est de trouver tous les nombres premiers jusqu'à un certain rang. Pour cela il procède en deux phases :

- Se placer sur le prochain élément présent dans le tableau,
 - Enlever dans le reste de ce tableau tous les multiples de ce nombre.
- Ecrivez cet algorithme.

2 Tris simple

2.1 tri naïf

Réaliser un algorithme qui ordonne les éléments d'un tableau par ordre croissant.

2.2 tri radix (simplifié)

Pour cet algorithme de tri, on suppose que les nombres à trier sont compris entre 0 et 9. On se base alors sur un autre principe. Au lieu de comparer les valeurs deux à deux comme précédemment, on va remplir un autre tableau qui comptera les occurrences de chaque élément. Reste ensuite à parcourir ce tableau en réécrivant le bon nombre de valeurs.

exemple :

1	2	3	6	5	8	7	9	9	6	3	5	7	5	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

le tableau des occurrences :

valeurs	0	1	2	3	4	5	6	7	8	9
occurrences	0	1	2	2	0	3	2	2	1	2

le tableau trié :

1	2	2	3	3	5	5	5	6	6	7	7	8	9	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2.3 fusion de tableaux

Ecrire un algorithme qui, à partir de deux tableaux triés, remplit un troisième, trié lui aussi.

3 carré magique

Ecrivez un algorithme qui vérifie si un carré est un carré magique (de taille quelconque).

Rappel : un carré est un carré magique si la somme de chacune des lignes = somme de chacune des colonnes = somme des diagonales.

4 Matrices

On travaille ici sur des matrices à deux dimensions.

4.1 Somme

Ecrivez un algorithme qui réalise la somme de deux matrices $M1$ et $M2$ dans une matrice $M3$. $M3(i, j) = M1(i, j) + M2(i, j)$. On suppose $M1$ et $M2$ de même dimensions.

4.2 Transposé

Ecrivez un algorithme qui réalise le transposé d'une matrice $M1$. $M2(i, j) = M1(j, i)$

4.3 Produit

Ecrivez un algorithme qui réalise le produit de deux matrices $M1$ et $M2$ dans une matrice $M3$. $M3(i, j) = \sum_{k=1}^n M1(k, j) * M2(i, k)$

5 n-reines

Au échecs, la reine peut se déplacer selon les lignes, les colonnes et les diagonales, d'autant de cases qu'elle veut. Le problème des n-reines consiste

a placer n reines sur un échiquier de taille $n * n$ sans qu'aucune ne puisse prendre. On suppose disposer d'une configuration des reines sur l'échiquier :

Ecrivez un algorithme qui vérifie si une solution au problème des n reines est valide.