

ESI 2011/2012 – 2e année CPI – Structures de Fichiers (SFSD)
Examen Semestriel du 25/01/2012 – Doc. Interdit – Durée 2h

Chaque exercice doit être traité sur une double feuille à part.

1) Tableaux volumineux en MS

Dans ce qui suit, on représente les tableaux d'entiers par des fichiers TÔF, car ils ne peuvent pas être chargés entièrement en mémoire centrale.

Les éléments d'un tableau de taille n , sont stockés séquentiellement dans le fichier. La capacité des blocs est de b entiers. Tous les blocs sont donc forcément remplis à 100% sauf le dernier.

- a) Quels sont les caractéristiques d'un tel fichier et donnez sa déclaration.
- b) Donnez un algorithme efficace qui trouve la valeur moyenne des éléments d'un tableau.
- c) Donnez un algorithme efficace qui réalise la multiplication de deux tableaux comme suit :
 $(a_1, a_2, \dots, a_n) \times (b_1, b_2, \dots, b_n) = (a_1 * b_1, a_2 * b_2, \dots, a_n * b_n)$
- d) Questions de cours :
 - Quel est le coût des opérations de recherche et d'insertion dans un fichier non ordonné et indexé ?
 - Quels sont les avantages et inconvénients des fichiers TOF ?

2) Matrices volumineuses en MS

Dans ce qui suit, on représente des matrices d'entiers (tableaux à deux dimensions) par des fichiers TÔF, car elles ne peuvent pas être chargées entièrement en mémoire centrale.

Les éléments d'une matrice de n lignes et m colonnes, sont stockés séquentiellement, ligne par ligne dans le fichier (c'est à dire que les éléments de la ligne i sont stockés avant ceux de la ligne $i+1$ et pour une même ligne, l'élément de colonne j précède celui de la colonne $j+1$). La capacité des blocs est de b entiers. Tous les blocs sont donc forcément remplis à 100% sauf le dernier.

- a) Quels sont les caractéristiques d'un tel fichier et donnez sa déclaration.
- b) Donnez un algorithme efficace qui trouve l'adresse (numéro de bloc et numéro d'enregistrement) de l'élément m_{ij} (qui se trouve à la ligne i et colonne j) d'une matrice M .
- c) Donnez un algorithme efficace qui transforme une matrice $M1(n,m)$ représentée ligne par ligne en une nouvelle matrice $M2(n,m)$ représentée colonne par colonne (c'est à dire que les éléments de la colonne j sont stockés avant ceux de la colonne $j+1$ et pour une même colonne, l'élément de la ligne i précède celui de la ligne $i+1$).
- d) Questions de cours :
 - Est-il possible d'obtenir un chargement proche de 100% après un chargement initial dans un B-Arbre ?
 - Quelle est l'utilité de garder les clés primaires, au lieu des adresses vers le fichier de données au niveau des listes inversées dans les méthodes d'index secondaires.