IN 101 - TD 07
5 novembre 2010

Énoncé

Le but du TD est la manipulation de matrices. On souhaite créer ces matrices en mémoire et que les différentes fonctions y accèdent. Les matrices seront stockées sous la forme d'un tableau des lignes de type `int **` et chaque case contiendra une ligne de type `int *`, c'est-à-dire un pointeur sur la première case de la ligne.

Exercice 1 : Petits Exercices sur les pointeurs

1.a] Écrire un programme qui déclare deux entiers `a` et `b`, initialisés par des valeurs prises sur la ligne de commande, ainsi que deux pointeurs `p` et `q`. Le pointeur `p` est défini pour pointer sur `a`, tandis que `q` est défini pour pointer vers un nouvel entier alloué dynamiquement. Ce dernier prend la valeur de `b` ajoutée à celle de pointée par `p`. Afficher les valeurs de `a`, `b`, `p` et `q` avant initialisation, puis après m'initialisation leur valeur ainsi que les valeurs de `*p` et `*q` et les adresses de p et q éventuellement. (À la place de `*p`, écrire `p[0]`.)

1.b] Écrire une fonction `void echange(int a, int b)` qui devrait échanger des valeurs sans utiliser de pointeurs. La corriger pour quelle fonctionne bien en utilisant un passage d'argument par adresse.

Exercice 2 : Gestion des matrices

2.a] Écrire une fonction qui réserve la mémoire et initialise les cases avec des valeurs aléatoires. (Reprendre le TD sur les tris pour l'initialisation aléatoire.)

2.b] Écrire une fonction qui affiche une matrice.

Exercice 3 : Quelques opérations

3.a] Écrire une fonction qui réalise l'addition de deux matrices et tester la.

3.b] Écrire une fonction qui réalise la multiplication de deux matrices et tester la.

Exercice 4 : Calcul du déterminant

4.a] Calculer le déterminant d'une matrice en utilisant un algorithme récursif. On rappelle la formule de Lagrange : $det(M) = \sum_i (-1)^{i+j} M_{i,j} det(M_j^i)$ où M_i^j est la sous-matrice obtenue en supprimant la i-ième ligne et la j-ième colonne.

(Utiliser un développement suivant la dernière colonne. Sauver la matrice avant de calculer la sous-matrice. Utiliser le fait que votre tableau de lignes permet de manipuler des lignes directement. La comatrice pour a[n − 1][n − 1] consiste simplement à enlever la dernière ligne et la dernière colonne. On n'a pas besoin d'enlever les valeurs sur la dernière colonne, il suffit de dire que la longueur de la ligne est `n-1`. Ensuite, la comatrice du coefficient a[n − 2][n − 1], il faut déplacer la dernière ligne à la place de l'avant-dernière ligne, ce qui se fait bien en manipulant simplement la ligne. Est-ce nécessaire de copier toute la matrice ou seulement le tableau de lignes ?)

4.b] Quelle est la complexité de votre algorithme de calcul de déterminant. (L'algorithme de Gauss qui met sous une forme triangulaire supérieure une matrice a une complexité en n^3 opérations où n est la taille de la matrice. La récursivité n'est pas toujours une solution efficace !)

Exercice 5 : Multiplication de Strassen (Groupe fort)

5.a] Quelle est la complexité de l'algorithme naïf de multiplication de deux matrices de taille $n \times n$ en nombre d'additions et de multiplications d'entiers ? Peut-on faire moins de n^3 multiplications d'entiers ?

$$\begin{pmatrix} r & s \\ t & u \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e & f \\ g & h \end{pmatrix}$$

$$\begin{aligned} p_1 &= a(g-h) \\ p_2 &= (a+b)h \\ p_3 &= (c+d)e \\ p_4 &= d(f-e) \\ p_5 &= (a+d)(e+h) \\ p_6 &= (b-d)(f+h) \\ p_7 &= (a-c)(e+g) \end{aligned}$$

On peut écrire :

$$\begin{aligned} r &= p_5 + p_4 - p_2 + p_6 \\ s &= p_1 + p_2 \\ t &= p_3 + p_4 \\ u &= p_5 + p_1 - p_3 - p_7 \end{aligned}$$

5.b] Calculer le nombre d'addition et de multiplication dans le cas de l'algorithme de multiplication naïf et avec celui de Strassen pour des matrices 2×2 ?

5.c] Supposons que $n = 2m$, décomposer la matrice initiale en 4 blocs de matrices $m \times m$ et appliquons une fois Strassen à la fin. Calculer le nombre de multiplications et d'additions d'entiers en fonction de n ?

5.d] Supposons n une puissance de 2. On peut toujours rajouter des zéros. Soit $M(n)$ le nombre de multiplication réalisées par Strassen pour multiplier 2 matrices de taille n et $A(n)$ pour le nombre d'additions. Écrire la formule de récurrence sur le nombre de multiplications et le nombre d'addition en fonction de n, et résoudre.

5.e] S'il vous reste du temps, implémentez l'algorithme de Gauss, calcul du rang et déterminant d'une matrice, et l'inverse.