

# Contrôle - Décembre 2014

Durée : 1h30

Les notes de cours, de travaux dirigés et de travaux pratiques sont autorisées. Le sujet comporte deux pages et deux parties (première partie : 10 points, deuxième partie : 10 points). Vous vous attacherez à soigner la présentation générale et l'orthographe, et vous serez notamment vigilant quant à la syntaxe lors de l'écriture de fragment de code.

Le sujet vous impose de programmer en utilisant des classes et des méthodes. Vous devez respecter cette contrainte !

## 1 Matrices de Toeplitz (10 pts)

On considère une classe particulière de matrices carrées, les matrices de Toeplitz. Ces matrices sont caractérisées par le fait que les coefficients sur une diagonale descendant de gauche à droite sont les mêmes. Par exemple, la matrice (carrée) suivante  $M$  vérifie cette propriété :

$$M = \begin{pmatrix} a & b & c & d & e \\ f & a & b & c & d \\ g & f & a & b & c \\ h & g & f & a & b \\ j & h & g & f & a \end{pmatrix}$$

**Definition** Etant donnée une matrice  $M$  carrée  $n \times n$ ,  $M$  sera dite de Toeplitz si pour tout  $k$  et  $l$   $1 \leq l, k \leq n - 1$ , on a  $M_{k,l} = M_{k-1,l-1}$ .

On pourrait décrire une telle matrice de taille  $n \times n$  à l'aide d'un tableau de taille  $n^2$  (On le fera d'ailleurs dans la partie 2). Cependant, dans cette partie, on choisit une représentation permettant de ne stocker qu'une seule fois chacun des coefficients.

**Question 1** Dans l'exemple d'une matrice  $5 \times 5$  (voir figure), expliquez pourquoi il suffit de stocker 9 coefficients pour reconstruire la matrice  $M$ . En déduire la taille minimale nécessaire pour coder complètement une matrice de Toeplitz  $n \times n$  dans un tableau.

**Question 2** Proposez une structure de données pour représenter de manière compacte une matrice de Toeplitz de taille  $n \times n$  avec  $n$  quelconque. Comment serait représentée la matrice ci-dessus avec cette structure de données ?

**Question 3** Ecrivez un constructeur permettant de construire la matrice nulle (qui est une matrice de Toeplitz).

**Question 4** Proposez une opération d'addition de 2 matrices de Toeplitz.

**Question 5** Proposez une opération d'affichage sous forme d'un tableau de taille  $n \times n$  d'une matrice de Toeplitz de taille  $n \times n$ .

## 2 Changements de représentation des matrices (10 pts)

Habituellement, on représente les matrices de taille  $n \times n$  par un tableau de taille  $n^2$  où les coefficients sont rangés ligne par ligne.

**Question 6** Indiquez la structure de données classique à utiliser pour représenter les matrices carrées de taille  $n \times n$  de cette manière.

**Question 7** Proposez une fonction qui teste si une matrice  $n \times n$  décrite dans cette structure de données est bien une matrice de Toeplitz.

**Question 8** Proposez une opération de traduction d'une matrice de Toeplitz vers une matrice représentée par ses  $n^2$  coefficients. Pourquoi est-il nécessaire d'utiliser la méthode `set` de la classe `matrice` pour pouvoir effectuer cette opération ?

**Question 9** Proposez une opération de traduction d'une matrice codée classiquement vers le représentation compacte des matrices de Toeplitz. Que faut-il vérifier avant de procéder à la traduction ?

**Question 10** On suppose que l'on dispose de l'opération de multiplication sur la représentation classique des matrices. Proposez un moyen de faire le produit de deux matrices de Toeplitz. Dans quelle structure de données doit-on stocker le résultat ?

## A Classes à utiliser

```
class matrice;
class toeplitz;

class matrice
{
    int *t;
    int size;

public:
    matrice(int n);
    void set(int i, int j, int v); // stocke la valeur v dans la case i,j de la matrice
    toeplitz from_matrice();
    bool is_toeplitz();
    //matrice operator+(matrice t2);
    //matrice operator*(matrice t2);
    friend ostream& operator<<(ostream &o, matrice t);
};

class toeplitz
{
    int *t;
    int size;

public:
    toeplitz(int n);
    void set(int p, int v);
    toeplitz operator+(toeplitz t2);
    matrice from_toeplitz();
    friend ostream& operator<<(ostream &o, toeplitz t);
    friend istream &operator>>(istream &i, toeplitz &t);
};
```