

DEVOIR EN TEMPS LIBRE D'INFORMATIQUE N° 4

Vous numéroterez vos copies et ferez apparaître clairement sur la première page le nombre de copies. Vous prêterez une attention particulière **au soin** de vos copies.

Si vous avez accès à un ordinateur muni de Python (environnement de base ou plus évolué), n'hésitez pas à tester vos fonctions.

1 Exercice : Travail sur les matrices

Dans cet exercice, on considère que les matrices seront représentées par des tableaux bidimensionnels (listes de listes). Par exemple le tableau LM valant `[[1,2,3],[0,1,5]]` représente la matrice $A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 5 \end{pmatrix}$

Attention : le terme `LM[i][j]` représente le coefficient $a_{(i+1),(j+1)}$ de la matrice A.

Par exemple le coefficient 5 de la matrice précédente est obtenu par la syntaxe `LM[1][2]`

1. Dimensions d'une matrice

Ecrire une suite d'instructions qui, étant donnée une liste LM représentant une matrice **M**, affecte à la variable **haut** la hauteur de **M** et à la variable **larg** la largeur de **M**.

2. Copie d'une matrice

Ecrire une suite d'instructions qui, étant donnée une liste LM représentant une matrice **M** (*de taille inconnue*), affecte à la variable **LMbis** une copie de la matrice **M**. Attention : On ne veut pas qu'après cette copie, un changement des coefficients de la liste LM affecte ceux de la variable **LMbis** et réciproquement.

3. Produit et transposée de matrices

- (a) Ecrire une fonction de nom **transpose** qui reçoit comme argument une liste LM représentant une matrice M (*de taille inconnue*) et qui retourne la liste représentant la transposée de la matrice M.
- (b) Ecrire une fonction de nom **produitpossible** qui reçoit comme arguments deux listes LM et LN représentant les matrices M et N (*de tailles inconnues*) et qui retourne le booléen True si les dimensions sont compatibles avec le calcul du produit MN et False sinon.
- (c) Ecrire une fonction de nom **produit** qui reçoit comme arguments deux listes LM et LN représentant les matrices M et N (*de tailles inconnues*) et qui retourne la liste représentant le produit MN si les dimensions sont compatibles avec le calcul du produit MN, et qui retourne un message d'erreur sinon. Calculer le nombre d'additions, de multiplications et d'affectations effectuées en fonction des tailles des deux matrices M et N.

4. Opérations élémentaires

- (a) Ecrire une fonction de nom **dilatation_ligne** qui reçoit comme arguments une liste **LM** représentant une matrice M (*de taille inconnue*), un entier naturel **p** et un nombre **mu**, et qui retourne le tableau correspondant à la matrice obtenue à partir de M après dilatation de la ligne L_p de rapport **mu**. Vous prêterez attention au fait que la ligne L_p de la matrice est constituée des coefficients de la liste **LM[p-1]**.

Vous écrirez trois versions de cette fonction :

- i. une version, notée provisoirement **dilatation_ligne_prod**, en utilisant la fonction **produit** et en introduisant une matrice de dilatation

- ii. une version, notée provisoirement **dilatation_ligne_cop**, en créant une copie de la liste représentant la matrice et en n'effectuant que les changements correspondant à la ligne L_p de la matrice
 - iii. une version, notée provisoirement **dilatation_ligne_bord**, qui n'a pas de retour mais qui transforme la liste **LM** directement en n'effectuant que les changements correspondant à la ligne L_p de la matrice.
 - iv. Comparer les complexités de ces trois fonctions en nombre d'additions, de multiplications, d'affectations ou de copies de nombres entiers ou flottants, en fonction de la taille de la matrice entrée en argument.
- (b) Ecrire une fonction de nom **transvection_ligne** qui reçoit comme arguments une liste **LM** représentant une matrice M (*de taille inconnue*), deux entiers naturels **k** et **j** (*supposés distincts et compatibles avec la taille de la matrice*) et un nombre **mu**, et qui retourne le tableau correspondant à la matrice obtenue à partir de M après la transvection : $L_k \leftarrow L_k + mu L_j$. On n'écrira qu'une version de cette fonction mais on en donnera la complexité.
- (c) Ecrire une fonction de nom **transposition_ligne** qui reçoit comme arguments une liste **LM** représentant une matrice M (*de taille inconnue*) et deux entiers naturels **k** et **j** (*supposés distincts et compatibles avec la taille de la matrice*), et qui retourne le tableau correspondant à la matrice obtenue à partir de M après la transposition : $L_k \longleftrightarrow L_j$. On n'écrira qu'une version de cette fonction mais on en donnera la complexité.