Lycée Loges

Chapitre 8

Manipulations d'images Exercices

Simon Dauguet
simon.dauguet@gmail.com

13 mars 2025

1 Manipulations matricielles

Exercice 1 :

1. Préciser la valeur de la matrice `m = 4 * [0] * 6`.
2. On définit la matrice `m = [ 5*[0] ] * 3`.
   (a) Écrire la valeur de `m`.
   (b) Que vaut la matrice `m` après l'instruction `m[0][1] = 1` ? Pourquoi ?
   (c) Quelle instruction par compréhension faut-il faire pour définir correctement la matrice `m` ?

Exercice 2 :
Définir une fonction `matVandermonde(tabA:list, n:int) -> list`, qui renvoie la matrice de Vandermonde associée à `tabA=[a1, a2, ..., am]`, à savoir $\begin{pmatrix} 1 & a_1 & a_1^2 & \dots & a_1^n \\ 1 & a_2 & a_2^2 & \dots & a_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & a_m & a_m^2 & \dots & a_m^n \end{pmatrix}$.

Exercice 3 :
Proposer une fonction `det2(mat:list) -> float`, qui renvoie le déterminant de `mat` matrice de $\mathcal{M}_2(\mathbb{R})$.

Exercice 4 :
Comme pour les vecteurs, multiplier une matrice par un scalaire consiste à multiplier tous les coefficients matriciels par ce scalaire, ainsi $3 \times \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 3 & 6 & 9 \\ 12 & 15 & 18 \end{pmatrix}$.

1. Proposer la fonction `pMatrice(mat:list, k:float) -> list`, qui renvoie le produit $k \times mat$.
2. Préciser la complexité temporelle selon les dimensions de $mat \in \mathcal{M}_{n,m}$.

Exercice 5 (Matrice stochastique) :
Une matrice est dite stochastique (ou probabiliste) lorsqu'elle est carrée, que tous ces coefficients sont positifs et que pour chacune des lignes la somme de ces coefficients vaut un.

Par exemple, si $M_1 = \begin{pmatrix} 0 & 1 \\ 1/2 & 1/2 \end{pmatrix}$, $M_2 = \begin{pmatrix} 0 & 1/10 \\ 1/2 & 1/2 \end{pmatrix}$, $M_3 = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 1/4 & 1/4 \end{pmatrix}$ et $M_4 = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & -1/2 & 1 \\ 0 & 2/3 & 1/3 \end{pmatrix}$.

Seule M_1 est une matrice stochastique (M_2 possède une somme distincte de 1, M_3 n'est pas carrée, M_4 possède un coefficient strictement négatif).

1. Proposer une fonction `estStochastique(mat:list) -> bool`, qui précise si la matrice fournie est stochastique.
2. Une matrice est bistochastique si elle est stochastique et en plus la somme des coefficients de chacune des colonnes vaut un. Proposer une fonction `estBiStochastique(mat:list) -> bool`.

Exercice 6 (Produit de Hadamard) :
Le produit matriciel de Hadamard (utiliser dans la compréhension JPEG) de deux matrices A et B est le produit de chaque coefficient $A_{i,j}$ avec $B_{i,j}$.

Par exemple, si $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$, et $B = \begin{pmatrix} 5 & 6 \\ 9 & 0 \end{pmatrix}$ alors $A \cdot B = \begin{pmatrix} 5 & 12 \\ 27 & 0 \end{pmatrix}$.

1. Proposer une fonction `prodHadamard(mat1:list, mat2:list) -> list`, qui renvoie le produit matriciel de Hadamard après avoir vérifié l'égalité des tailles des matrices.
2. Préciser la complexité temporelle si les deux matrices sont dans $\mathcal{M}_{n,m}$.

Exercice 7 (Produit matricielle) :
Proposer une fonction `prodMat(mat1:list, mat2:list) -> list` qui effectue le produit matriciel (le vrai) après avoir vérifié la taille des matrices.

2 Manipulations d'images

On supposera les images déjà traitées et donc, les images seront considérés comme des tableaux de nombres (*i.e.* les fonction de transformations `photo2matrice` seront déjà utilisée).

Exercice 8 :
Proposer une fonction `grisMoyen(mat:list) -> int`, qui renvoie la valeur moyenne (arrondie à l'entier) des pixels de `mat`. Préciser la complexité temporelle selon les dimensions de $mat \in \mathcal{M}_{n,m}$.

Exercice 9 :
Proposer une fonction `couleursMoyennes(mat:list) -> tuple`, qui renvoie le tuple des valeurs moyennes (arrondies à l'entier) de chaque couleur RGB des pixels de `mat`.

Exercice 10 :
Proposer une fonction `filtreSeuil(mat:list, s:int) -> list`, qui renvoie la matrice dont les coefficients (i,j) valent 255 si $mat_{i,j} > s$, 0 sinon.

Exercice 11 :

Un apprenti espion a encodé une image à l'aide de la fonction f qui à x renvoie $93x\ [256]$ (le reste de la division euclidienne de $93x$ par 256). Pour annuler ce codage, il faut trouver un entier k tel que le reste de la division du produit $k \times 93$ par 256 fasse 1, puis faire la même manipulation.

1. Déterminer $k \in \mathbb{N}$ tel que $93k \equiv 1\ [256]$
2. Écrire une fonction `antiEspion(mat:list, k:int) -> list`, qui renvoie la matrice dont les coefficients (i, j) valent le reste de la division euclidienne par 256 de $k \times mat_{i,j}$.
3. Déterminer alors l'image qui se cache derrière `cpge-goldenEspion.bmp`.

Exercice 12 :

On se propose d'encoder la rotation d'angle $+\frac{\pi}{2}$ à partir du coin en haut à gauche d'une image et on considère la matrice suivante $M = \begin{pmatrix} 0 & 0 & 255 & 255 & 255 \\ 255 & 0 & 255 & 255 & 255 \\ 255 & 0 & 0 & 0 & 0 \\ 255 & 255 & 0 & 0 & 255 \end{pmatrix}$.

1. Faire une rotation de $+\frac{\pi}{2}$ à la matrice M.
2. Où la case (0,0) est-elle envoyée ? La case (2,0) ? La case (0,3) ? La case (2,1) ? La case (2,2) ?
3. Proposer les coordonnées de la nouvelle case en fonction des coordonnées de l'ancienne case (i, j).
4. Proposer une fonction `rotation(mat:list) -> list`, qui permette d'obtenir une telle rotation d'angle $+\frac{\pi}{2}$, puis l'appliquer à l'image `cpge-champigris.bmp`.
5. Comment faire pour une rotation de $-\frac{\pi}{2}$?

Exercice 13 :

On souhaite réduire la taille d'une image par quatre. L'algorithme consiste à remplacer quatre pixels contiguës par un seul pixel de valeur moyenne des quatre. Proposer une fonction `reduction4(mat:list) -> list`, qui renvoie une matrice réduite répondant au problème. Tester sur une image.

Exercice 14 :

Proposer une fonction `composantes(mat:list) -> tuple`, qui renvoie à partir d'une matrice représentante une image couleurs, le tuple des trois matrices de chacune des composantes : rouge, verte et bleue.

exemple : si $mat = \begin{pmatrix} (0, 120, 120) & (50, 170, 180) \end{pmatrix}$ alors le tuple contiendra les trois matrices $\begin{pmatrix} (0, 0, 0) & (50, 0, 0) \end{pmatrix}$, $\begin{pmatrix} (0, 120, 0) & (0, 170, 0) \end{pmatrix}$ et $\begin{pmatrix} (0, 0, 120) & (0, 0, 180) \end{pmatrix}$.

En enregistrant les trois résultats sous forme d'image, tester sur une image couleur de votre choix.

Exercice 15 :

Proposer une fonction `reduction4C(mat:list) -> list`, qui réduise par quatre l'image couleur représentée par la matrice `mat`. Tester sur une image.