

Matrices et Images

1 Listes de listes

1.1 Introduction aux listes de listes

On considère la liste suivante :

```
L=[[1,2,3,4],["poireau","carotte","patate"],[5,6,7,8,9]]
```

1. Recopier cette liste dans la console. Essayez de prévoir ce que valent `L[2]` ; `len(L)` ; `len(L[0])` ; `len(L[1])` et `L[1][1]`. Vérifier vos prédictions à la console.
2. Écrire une fonction `testelt(LL,x)` qui prend en argument une liste de listes `LL` et une variable `x` (dont le type n'est pas spécifié) et qui renvoie un booléen indiquant si l'élément `x` est présent dans `LL`
3. Écrire une fonction `sommesouslistes(LL)` qui prend en argument une liste de listes `LL` dont les éléments sont des nombres et qui renvoie une liste `L` où chaque `L[i]` est la somme des éléments de la liste `LL[i]`

1.2 Matrices vues comme liste de listes

1. Écrire une fonction qui teste si une liste de listes est susceptible de représenter une matrice rectangulaire. Cette fonction renverra un booléen.
2. Écrire une fonction qui teste si une liste de listes est susceptible de représenter une matrice carrée.
3. Créer sous forme de listes de listes la matrice `M` et le vecteur colonne `X` :

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad \text{et} \quad X = \begin{pmatrix} 10 \\ 11 \\ 12 \end{pmatrix}$$

4. Écrire une fonction `prodmatvcol(LL,X)` qui prend en argument une matrice `LL` carrée d'ordre `n` (représentée par une liste de listes) et un vecteur colonne `X` et qui renvoie le vecteur colonne `LL*X`
5. Vérifier votre fonction avec les matrices `M` et `X` définies plus haut.

2 Type array de numpy

Le module `numpy` est spécialisé dans le calcul sur les listes et les matrices. Il propose plusieurs types matriciels ; nous nous servirons du plus simple dans ce TP, à savoir le **tableau numpy** nommé en anglais (et en Python) `np.array`

1. Après avoir importé `numpy` convertir les listes de listes `M` et `X` précédentes en tableaux numpy à l'aide d'une syntaxe de la forme `A=np.array(A)`
2. Le produit matriciel de `A` par `B` avec des tableaux numpy s'écrit `np.dot(A,B)`. Vérifier à l'aide de cette syntaxe les résultats de la première partie.
3. **Exo étoile** Que produit la commande `np.shape(M)` ? Et la commande `M.append([10,11,12])` ? Que produit `T=np.zeros((4,4))` ? Proposer alors une fonction qui prend en argument une matrice `A` quelconque et renvoie la matrice construite par blocs :

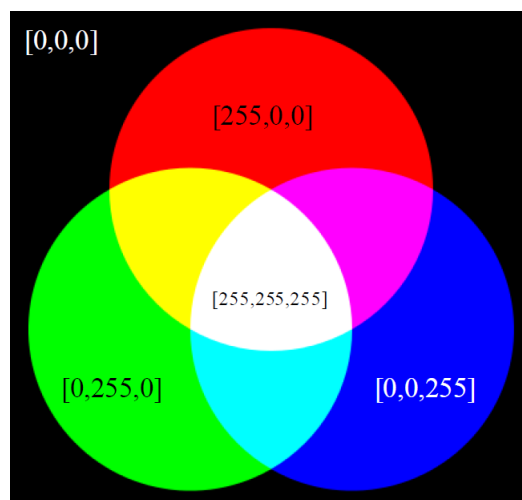
$$\begin{pmatrix} A & 3A \\ 2A & 4A \end{pmatrix}$$

3 Images

Une image est formée de pixel (« picture element ») : un pixel est un carré de la plus petite dimension affichable par l'écran (ou projecteur) considéré, au quel on affecte une couleur. A chaque pixel est associé une information :

- booléenne pour une image en noir ou blanc ;
- entier naturel pour une image en nuance de gris (compris entre 0 et 255) ;
- 3 entiers naturels (codés dans une liste) pour une image en couleur.

Pour définir la couleur d'un pixel, nous utiliserons le codage le plus répandu, connu sous le nom de codage RGB. Il faut préciser les intensités des trois couleurs primaires qui formeront la couleur du pixel : le premier nombre précise l'intensité du rouge, le second celle du vert et le troisième celle du bleu.



Les fichiers suivants sont fournis :

- Gall-Peters_projection.jpg : une image de la terre en projection selon Gall-Peters
- image.py : un programme à compléter pour réaliser les fonctions demandées

Ouvrir le fichier image.py

1. Expliquer ce qui y est fait lors des 8 premières lignes du programme.
2. Quelles sont les dimensions de l'image (horizontalement et verticalement)? Quel est le type colorimétrique de l'image? Quelles sont les dimensions du tableau correspondant? Afficher dans la console terretab.
3. Justifier le type de valeurs utilisé dans le tableau.
4. Déterminer le profil colorimétrique du pixel en position 125 horizontalement et en position 250 verticalement.

On cherche à faire tourner cette image de $+90^\circ$

On va donc créer un nouveau tableau terretabnew qui sera converti en image par les deux dernières lignes du programme.

5. Supprimer les marques de commentaire # du programme. A la vingtième ligne, indiquer entre crochet les dimensions du nouveau tableau.
6. En utilisant des boucles, construire un nouveau tableau correspondant à l'image rechercher.

4 Exo étoile : produit matriciel

1. Écrire une fonction qui prend en arguments deux listes de listes représentant deux matrices M1 et M2 et qui vérifie que les dimensions de ces matrices sont compatibles avec le calcul du produit $M1 * M2$
2. Écrire une fonction qui prend en arguments deux listes de listes représentant deux matrices M1 et M2 compatibles avec le produit matriciel et qui renvoie la matrice produit $M1 * M2$
3. Quelle est la complexité de la fonction précédente ?
4. Écrire une fonction qui calcule M^n , avec M matrice d'ordre carré m et n un entier strictement positif. Quelle est la complexité de ce calcul ?
5. Réfléchir à comment on pourrait améliorer la complexité de ce calcul !