

# Méthodologie de la Programmation

TD - 2 Juin 2017

## Matrices

Une **matrice** est un tableau de nombres à 2 dimensions, qui peut se représenter en python par une liste de listes, comme par exemple :

```
M = [[1, 4, 7, 5],  
      [8, 2, 3, 0],  
      [9, 7, 5, 4]]
```

⇒ Pour accéder à  $M_{ij}$  (élément à la ligne  $i$  et colonne  $j$ ), on écrira simplement `M[i][j]`.

## Transposée

La **transposée** d'une matrice est obtenue en transformant les lignes en colonnes et vice-versa, par exemple pour M on aura la matrice transposée  $M^T$  :

```
Mt = [[1, 8, 9],  
       [4, 2, 7],  
       [7, 3, 5],  
       [5, 0, 4]]
```

## Matrices creuses

Dans le cas où la matrice est composée d'un grand nombre de zéros, on parle de matrice **creuse**. Afin d'éviter le stockage des zéros d'une telle matrice, différents formats sont utilisés pour ne stocker que les valeurs non-zéro. L'un des plus connus consiste à stocker l'index de la colonne correspondant à chaque valeur différente de zéro.

Par exemple, cette matrice (creuse) :

```
[[8, 0, 5, 0, 0, 0, 9, 0, 0, 8],  
 [0, 0, 9, 0, 6, 0, 0, 0, 2, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 4],  
 [1, 8, 0, 2, 1, 0, 0, 0, 0, 0],  
 [0, 7, 0, 0, 0, 3, 0, 0, 0, 0]]
```

peut se représenter textuellement comme ceci (chaque ligne représente une nouvelle ligne dans la matrice) :

```
0:8 2:5 6:9 9:8  
2:9 4:6 8:2  
9:4  
0:1 1:8 3:2 4:1  
1:7 5:3
```

*Les exercices suivants sont indépendants.*

***Merci de rédiger dans un fichier .py portant votre nom, prénom et n. d'étudiant.***

***Vous devez envoyer votre travail sur le mail [josue.melka02@univ-paris8.fr](mailto:josue.melka02@univ-paris8.fr) à la fin de l'épreuve. Tout travail rendu après le délai fixé ne pourra être pris en compte.***

### Exercice 1: (4 points)

Pour une matrice quelconque de dimensions  $m \times n$  (m lignes et n colonnes), coder une fonction qui retourne la transposée de la matrice. Cette fonction aura pour signature :

```
def transpose(mat):  
    ...
```

### Exercice 2: (6 points)

A partir d'une matrice creuse de réels (type float) donnée sous sa forme pleine en python (liste de listes), coder une fonction permettant d'enregistrer cette matrice dans un fichier au format vu ci dessus. Cette fonction aura pour signature :

```
def write_sparse_matrix(mat, filename):  
    ...
```

### Exercice 3: (10 points)

A partir d'un fichier texte contenant une matrice creuse de réels (type float) au format vu précédemment, coder une fonction qui charge le contenu de ce fichier pour le restituer sous forme d'une matrice python (liste de listes). Cette fonction aura pour signature :

```
def read_sparse_matrix(filename):  
    ...
```