Exercice 1 – programmation dynamique (fin du TP 4)

En partant du sommet du triangle ci-dessous et en se déplaçant vers les nombres adjacents de la ligne inférieure, le total maximum que l'on peut obtenir pour relier le sommet à la base est égal à 23 :

$$\begin{array}{ccccccc} & & & \mathbf{3} & & & \\ & & \mathbf{7} & & 4 & & \\ & 2 & & \mathbf{4} & & 6 & \\ 8 & & 5 & & \mathbf{9} & & 3 \end{array} \qquad 3+7+4+9=23.$$

On modélise ce triangle à l'aide d'une liste de listes :

```
>>> t = [[3, 0, 0, 0], [7, 4, 0, 0], [2, 4, 6, 0], [8, 5, 9, 3]]
```

On dira par exemple que le coefficient à la ligne 2 et colonne 1 vaut 4.
Pour tout couple $(i, j) \in [\![0, n-1]\!]^2$, on note $S_{i,j}$ le plus grand total en partant du coefficient à la ligne i colonne j.

1. Que vaut $S_{n-1,j}$ pour $j \in [\![0, n-1]\!]$?
2. Donner une relation pour $0 \leqslant j \leqslant i \leqslant n-2$, une formule reliant $S_{i,j}$ avec $S_{i+1,j}$ et $S_{i+1,j+1}$.
3. Programmer une fonction `somme` prenant en argument un tel «triangle» et qui renvoie le total maximal pour un chemin reliant le sommet du triangle à sa base.

```
>>> t = [[3, 0, 0, 0], [7, 4, 0, 0], [2, 4, 6, 0], [8, 5, 9, 3]]
>>> somme(t)
23
```

4. Programmer ensuite une fonction `chemin` prenant elle aussi en argument un tel «triangle» et qui renvoie le couple formé par le total maximal et un chemin qui réalise ce total; on codera le chemin par la liste des colonnes visitées depuis la ligne 0 jusqu'à la ligne $n-1$.

```
>>> somme_bis(t)
(23, [0, 0, 1, 2])
```

Exercice 2 – complexité, diviser pour régner et utilisation de numpy

Dans cet exercice, on s'intéresse au produit matriciel dans $\mathscr{M}_n(\mathbb{R})$.

1. Si A et B sont deux matrices de $\mathscr{M}_n(\mathbb{R})$, combien de multiplications et d'additions sont effectuées avec l'algorithme usuel de produit de matrices.

2. Pourquoi est-il illusoire d'espérer un algorithme effectuant le produit de deux matrices de $\mathscr{M}_n(\mathbb{R})$ avec une complexité (correspondant au nombres de multiplications) de la forme $O(n^\alpha)$ avec $\alpha < 2$?

On suppose désormais avoir réalisé l'importation `import numpy as np` et l'on utilise des tableaux `numpy` pour représenter les matrices.

3. Écrire une fonction `blocs(M)` d'argument une matrice `M` de $\mathscr{M}_{2n}(\mathbb{R})$ et qui renvoie le tuple `A, B, C, D` de matrices de $\mathscr{M}_n(\mathbb{R})$ tel que l'on ait l'écriture par blocs $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$.

4. Écrire une fonction `assemble(A, B, C, D)` d'arguments quatre matrices carrées de même taille et faisant la démarche inverse de la question précédente en renvoyant la matrice $\begin{pmatrix} \mathtt{A} & \mathtt{B} \\ \mathtt{C} & \mathtt{D} \end{pmatrix}$.

En 1968, Strassen a remarqué la relation suivante pour des matrices de $\mathscr{M}_2(\mathbb{R})$:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} x_1 + x_2 - x_4 + x_6 & x_4 + x_5 \\ x_6 + x_7 & x_2 - x_3 + x_5 - x_7 \end{pmatrix},$$

avec :

$$x_1 = (b-d)(g+h)\ ,\ x_2 = (a+d)(e+h)\ ,\ x_3 = (a-c)(e+f)\ ,\ x_4 = (a+b)h\ ,$$

$$x_5 = a(f-h)\ ,\ x_6 = d(g-e) \text{ et } x_7 = (c+d)e.$$

Le produit des deux matrices de $\mathscr{M}_2(\mathbb{R})$ repose donc sur 7 multiplications et 18 additions.
On adopte alors une stratégie récursive en appliquant cette idée à un produit de deux matrices carrées à 2^n lignes et 2^n colonnes écrites en blocs :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \times \begin{pmatrix} E & F \\ G & H \end{pmatrix},$$

où A, B, C, D, E, F, G et H sont des matrices carrées à 2^{n-1} lignes et 2^{n-1} colonnes.
Le résultat de cette multiplication est une matrice de la forme :

$$\begin{pmatrix} X_1 + X_2 - X_4 + X_6 & X_4 + X_5 \\ X_6 + X_7 & X_2 - X_3 + X_5 - X_7 \end{pmatrix}$$

où $X_1, \ldots, X_7$ sont sept matrices à 2^{n-1} lignes et 2^{n-1} colonnes liées aux matrices A, B, C, D, E, F, G par des relations analogues à celles des matrices à 2 lignes et 2 colonnes.

5. On dispose de l'addition et de la soustraction `M+N` et `M-N` pour des matrices `M` et `N`.
Programmer une fonction récursive `produit(M, N)` où `M` et `N` sont des matrices carrées de taille une puissance de 2 et qui en calcule le produit à l'aide de l'algorithme ci-dessus.

6. On suppose que N est une puissance de 2. Combien y a-t-il de multiplications effectuées pour le calcul d'un produit de deux matrices de taille N par cet algorithme ?

7. On note $C(N)$ le nombre de multiplications, d'additions et de soustractions effectuées pour le calcul d'un produit de deux matrices de taille N par cet algorithme (où N est toujours une puissance de 2). Montrer que $C(N) = O\left(N^{\log_2(7)}\right)$.