INFORMATIQUE DS2

23 JANVIER 2017

EXERCICE 1 - SURVEILLANCE - TYPE INFO

L'ordre nouveau est arrivé, la population mondiale est surveillée par une intelligence artificielle supérieure dirigée par un consortium de professeurs de prépas de Blois.

Suivi des individus

Un ensemble d'individus est surveillé, et chacun reçoit une note (plus la note est grande, plus l'individu est dangereux, la note minimale est 0). Ces notes sont stockées dans une liste `notes` que l'on considère comme une variable globale. Chaque individu est déterminé par sa position dans la liste. Par exemple `notes[i]=k` signifie que l'individu i a une note k.

1. Écrire une fonction `trouver_les_leaders()` sans arguments qui renvoi la liste des individus ayant la note maximale.
2. Écrire une fonction `reformatage()` sans arguments qui réduit à 0 la note de ou des individus ayant la plus grande note.
3. Écrire une fonction `securisation(k)` qui prends en argument un entier k et réduit de k la note de tout les individus, avec un minimum de 0.

Risque de révolte

Pour endiguer une éventuelle révolte il convient d'analyser les risque en travaillant sur les notes.

4. Écrire une fonction `priorite()` qui renvoi le tableau `notes` trié dans l'ordre décroissant en effectuant un tri par insertion.
5. Justifier rapidement la complexité de ce tri.
6. Écrire une fonction `revolte(seuil)` qui renvoi un booléen selon la médiane des notes est supérieur ou égal à l'entier `seuil`. Par exemple si `notes=[5,3,1]`, `revolte(2)` renverra `True`.

Base pour le contrôle

Le contrôle de la population est fait grâce à une base SQL qui liste des délits commis par les individus. Cette base est constituée de deux tables :

La table `citoyens`, qui possède un enregistrement par personnes, dont les attributs sont :
- `matricule` le numéro de matricule de chaque individu, unique
- `nom` le nom de l'individu
- `profession` la profession de l'individu
- `note` la note de l'individu

La table `crimes`, qui possède un enregistrement par délits commis, dont les attributs sont :
- `id` un identifiant unique
- `mat` le numéro de matricule de l'individu en faute
- `type` le type de délits (`'vol'`, `'terrorisme,'cours non connu'`...)
- `date` au format jour-mois-année par exemple : '2015-05-17'

Écrire des requêtes SQL :

7. permettant d'obtenir les matricules des individus dont la profession est 'enseignant'
8. permettant d'obtenir les noms des individus ayant commis des délits le 22 juin 2013.
9. permettant d'obtenir les noms des individus ayant la note maximale.

Image d'un immeuble

Cette fois on s'intéresse à une population d'un immeuble et au risque de représente chaque foyer. Le but est de construire une image en noir et blanc pour détecter les zones de risques : le noir est une risque fort et le blanc un risque faible.

Les foyers sont repérés par leur coordonnées `(x,y)` de leur appartement. Les relevés sont situés dans un tableau `releve` ou `releve[x,y]=(n,P)` avec :

— `x` et `y` sont les coordonnées de l'appartement, avec $0 \leqslant x \leqslant 100$ et $0 \leqslant x \leqslant 80$

— `n` est un entier correspondant au nombre d'individus du foyer

— `P` est une liste de longueur `n` donnant le potentiel de dangerosité de chaque individu entre 0 et 1.

Une image en niveau de gris est un tableau de tableau `image` en Python. Pour simplifier la valeur `image[x][y]` de chaque pixel de l'image doit contenir un niveau de gris (entre 0 pour noir et 1 pour blanc) pour l'appartement `(x,y)`.
Une image en noir et blanc suit le même principe mais ne prends comme valeurs que des 0 ou des 1.
Le calcul du niveau de gris G se fait par en calculant en ôtant à 1 la moyenne du tableau `P`. Le niveau de bascule entre le niveau de gris et le noir ou blanc est de 0,6.

1. Écrire une fonction `image_gris(releve)` qui construit et renvoie cette image en niveau de gris.
2. Écrire une fonction `image_risques(releve)` qui construit et renvoie l'image des risques en noir et blanc.

EXERCICE 2 - SUITE VECTORIELLE - TYPE CCP MATHS

Un vecteur x de $\mathbb{R}^p$ sera représenté en Python par une liste de flottants; Par exemple le vecteur $x = (1,2,3)$ de $\mathbb{R}^3$ sera représenté par la liste `[1,2,3]`. De même une matrice A sera représenté par une liste dont les éléments sont les lignes de la matrice. Par exemple la matrice $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ sera représentée par la liste `[[1,2,3],[4,5,6]]`.

1. On exécute le script suivant `A=[[1,2,3],[4,5,6],[7,8,9],[10,11,12]]` qui représente la matrice $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$. Donner les valeurs renvoyées lorsque l'on exécute `len(A)`,`A[1]` et `A[2][1]`.
2. Écrire une fonction `difference` qui prends en arguments deux vecteurs x et y de même taille et renvoie le vecteur $x - y$. Par exemple si $x = (5,2)$ et $y = (3,7)$, `difference(x,y)` renverra `[2,-5]`.
3. Écrire une fonction `norme` qui prends en arguments un vecteur $x = (x_1, ..., x_p)$ et renvoie sa norme infinie $||x||_\infty = max\left\{|x_i| \, | i = 1..p\right\}$. (On pourra utiliser la fonction `abs` mais on s'interdit l'utilisation de la fonction `max`).
4. Écrire une fonction `itere` qui prends en arguments un vecteur ligne x et une matrice carrée A de même taille que x et qui renvoie le vecteur xA. Par exemple si $x = (1,1)$ et $A = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}$ on a $xA = (5,7)$ et donc `itere(x,A)` renverra `[5,7]`.
5. On s'intéresse à une suite de vecteurs lignes de $\mathbb{R}^p$ associée $(\mu_n)_{n\in\mathbb{N}}$ définie par la relation $\mu_{n+1} = \mu_n A$ ou A est une matrice de carrée de taille p. On peut démonter que sous certaines conditions sur la matrice A la suite $(\mu_n)_{n\in\mathbb{N}}$ converge vers un vecteur μ_∞.
Écrire une fonction `invariant` qui prends en arguments une matrice $A \in M_p(\mathbb{R})$ et un réel $\varepsilon > 0$ et qui renvoie le premier terme μ_k de la suite $(\mu_n)_{n\in\mathbb{N}}$ avec $\mu_0 = \left(\frac{1}{p}, \frac{1}{p}, ..., \frac{1}{p}\right)$ tel que $||\mu_k - \mu_{k-1}||_\infty \leqslant \varepsilon$. On ne se préoccupera pas des conditions sur A.

EXERCICE 3 - PHÉNOMÈNE PHYSIQUE - TYPE MODÉLISATION

On souhaite étudier un phénomène physique modélisé par l'équation différentielle $y''(t) = -\sin y(t) - t y'(t)^2$.
On souhaite résoudre cette équation différentielle numériquement via la méthode d'Euler.
On suppose connues les conditions initiales $y(0) = y_0$ et $y'(0) = z_0$.

1. On pose $z = y'$. Montrer alors que cette équation différentielle s'écrit sous la forme d'un système différentiel du premier ordre.
2. Une fonction `euler` est créée pour résoudre cette équation sur l'intervalle $[0,10]$. Recopiez et complétez cette fonction pour que les listes en sorties contiennent bien les valeurs des suites (y_i) et (z_i) correspondant aux approximations des fonctions y et z.

```
def euler(y0,z0,n):  # conditions initiales et nombre de points
    y,z,t = y0,z0,0 # valeurs initiales
    h = 10/n # pas
    Ly,Lz,Lt = [y],[z],[t] #  listes de stockage
    for in range(n-1) : # n points au total
        # à compléter
    return Ly,Lz,Lt
```

3. On considère qu'on a récupéré les valeurs dans les listes `Ly` et `Lz`. Proposer un code permettant de construire le portrait de phase de cette expérience.
4. On considère qu'on est en position d'équilibre lorsque $|y| < 10^{-6}$ et $|y'| < 10^{-6}$. Proposer une modification du code de la fonction `Euler` permettant de récupérer les valeurs jusqu'à la position d'équilibre.