L3 algèbre effective (12déc17) - Interrogation 2

Ex 1. On considère une famille $(u_{n,k})_{(n,k)\in\mathbb{N}\times\mathbb{N}}$ vérifiant les relations

$\sum_{k=0}^{n} u_{n,k} = 1$ n!

et

$\forall k > 0,\ u_{n,k} = \binom{n}{k} u_{n-k,0}$

La famille $(u_{n,k})$ est elle ainsi définie récursivement ?

Ecrivez un script calculant $u_{10,3}$. . (Le coefficient binomial peut être obtenu par l'instruction binomial(n,k).)

Ex 2. Ecrire une fonction inv(o) prenant comme argument une suite d'opérations o encodées comme dans les fonctions showop(o) et transf(A,o) définies plus bas et rendant la suite d'opérations telle que transf(transf(A,o),inv(o))=A pour toute matrice A.

Tester cette fonction. Que vaut transf(transf(A,inv(o)),o) ?

Ex 3. On considère la matrice A donnée par

```
A=matrix([\
[  2 ,  0 ,  1 , -4 ,  0 ,  6],\
[ -4 ,  2 ,  0 ,  8 , -2 ,-12],\
[  2 ,  6 ,  0 ,-10 , -6 ,  6],\
[  0 , -6 ,  0 ,  6 ,  6 ,  0]\
]);show('A=',A)
```

$$A=\begin{pmatrix} 2 & 0 & 1 & -4 & 0 & 6 \\ -4 & 2 & 0 & 8 & -2 & -12 \\ 2 & 6 & 0 & -10 & -6 & 6 \\ 0 & -6 & 0 & 6 & 6 & 0 \end{pmatrix}$$

Donner une listes d'opérations sur les lignes et les colonnes transformant A en sa "réduction de Smith".

Y a t-il des opérations inutiles dans la liste obtenue ?

Donner séparemment la liste des opérations sur les lignes et la liste des opérations sur les colonnes.

Construire des matrices P et Q telles que PAQ est nulle hors de la diagonale. Tester ce fait.

Exhiber une base du noyau de A (sous la forme de la matrice des coordonnées des vecteurs de la base). Le vecteur (3,2,0,0,2,-1) est il dans le noyau de A ? Si oui, quelles sont les coordonnées de ce vecteur dans la base choisie ?

Exhiber une base de l'image de A. Est-ce une base de $\mathbb{Z}^4$?

Ex. On considère la matrice $M = \begin{pmatrix} 3 & 0 \\ 0 & 6 \end{pmatrix}$ (M=matrix.diagonal([3,6]). Ecrire une fonction f(x,y) qui rend true (return(true)) si le couple d'entier (x,y) est dans l'image de M et rend false sinon.

Ecrire ensuite une fonction g(M,x,y), où M désigne une matrice 2x2 quelconque, qui rend true si (x,y) est dans l'image de M et qui rend false sinon. Tester cette fonction.

Pouvez vous généraliser la fonction g à une matrice M de taille 2xq avec q quelconque ?

Pouvez vous généraliser la fonction g à une matrice M de taille p x q et à un vecteur x de taille p ?

Fontions prédéfinies

```
def showop(o):
    l=""
    for op in o:
        if op[0]==1:l=l+"C"+str(op[1])+" + "+str(op[3])+"C"+str(op\
   [2])+" → C"+str(op[1])+"\n"
        elif op[0]==2:l=l+'L'+str(op[1])+' + '+str(op[3])+'*L'+str(\
   op[2])+' → L'+str(op[1])+"\n"
        elif op[0]==3:l=l+'C'+str(op[1])+"   C"+str(op[2])+"\n"
        elif op[0]==4:l=l+'L'+str(op[1])+"   L"+str(op[2])+"\n"
    return(l)
#usage : print showop(o)
```

```
import copy
def transf(A,o):
    B=copy.deepcopy(A)
    for op in o:
        if op[0]==1:B[:,op[1]-1]=B[:,op[1]-1]+op[3]*B[:,op[2]-1]
        elif op[0]==2:B[op[1]-1,:]=B[op[1]-1,:]+op[3]*B[op[2]-1,:]
        elif op[0]==3:
            C=B[:,op[1]-1];B[:,op[1]-1]=B[:,op[2]-1];B[:,op[2]-1]=C
        elif op[0]==4:
            L=B[op[1]-1,:];B[op[1]-1,:]=B[op[2]-1,:];B[op[2]-1,:]=L
    return(B)
```

```
def fsmith(A):
    p=A.nrows();q=A.ncols()
    L=[abs(A[i][j]) for i in range(p) for j in range(q)]
    if p*q==0 or max(L)==0:
        return([])
    else:
        a=min(L[i] for i in range(p*q) if L[i]!=0)
        k=L.index(a);i0=k//q;j0=k%q;a=A[i0][j0]
        l=[i for i in range(p*q) if L[i]%a!=0]
        if l==[]:
            o=[[1,j+1,j0+1,-A[i0][j]/a] for j in [0..q-1] if j!=j0]\
                + [[2,i+1,i0+1,-A[i][j0]/a] for i in [0..p-1] if i!=\
   i0]
            if j0!=0:o=o+[[3,1,j0+1]]
            if i0!=0:o=o+[[4,1,i0+1]]
            return(o+[[op[0],1+op[1],1+op[2]]+op[3:] for op in \
   fsmith(transf(A,o).submatrix(1,1))])
        else:
            k=min(l);i1=k//q;j1=k%q;b=A[i1][j1]
            if A[i0][j1]%a != 0:
                o=[[1,j1+1,j0+1,-(A[i0][j1]//a)]]
            elif A[i1][j0]%a != 0:
                o=[[2,i1+1,i0+1,-(A[i1][j0]//a)]]
            else:
                o=[[2,i1+1,i0+1,-(A[i1][j0]//a)],[2,i0,i1,1],\
                   [1,j1+1,j0+1,-(((1-A[i1][j0]//a)*A[i0][j1]+b)//a)\
   ]]
            return(o+fsmith(transf(A,o)))
```