Prática em matrizes (Python)

Esta folha tem como objetivo a prática em programação de matrizes. Para começar vai se ver como definir e manipular matrizes em Python. Para variar, há inumeras maneiras. Por exemplo, para definir uma matriz de 6 linhas por 6 colunas contendo zeros pode-se fazer

```
>>> a=[0]*6   # a é uma lista de 6 zeros
>>> b=[a]*6   # b é uma lista de 6 listas...
              # ou b é uma matriz de 6x6
```

```
ou por extenso:
>>> b=[[0,0,0,0,0,0],[0,0,0,0,0,0],
       [0,0,0,0,0,0],[0,0,0,0,0,0],
       [0,0,0,0,0,0],[0,0,0,0,0,0]]
```

```
ou usando o numpy:
>>> import numpy
>>> b=numpy.zeros((6,6))
```

Nas matrizes, define-se uma característica importante, denominada ordem. Vem a ser a quantidade de linhas, seguida pela quantidade de colunas da matriz. Assim, na matriz acima, de 6 linhas por 6 colunas, a ordem é o vetor `6,6`.

Uma matriz é quadrada quando o número de linhas é igual ao número de colunas. Diz-se neste caso que a ordem é um único número. No caso acima a matriz é quadrada de ordem `6`.

Pode-se criar uma função definida pelo usuário para criar matrizes. Ela poderia ter a seguinte definição:

```
def crie_mat(n_lin, n_col, val):
  mat = [] # lista vazia
  for i in range(n_lin):
    # cria a linha i
    lin = [] # lista vazia
    for j in range(n_col):
      lin.append(val)
    mat.append(lin)
  return(mat)
```

```
>>> crie_mat(3,4,5)
[[5, 5, 5, 5], [5, 5, 5, 5],
    [5, 5, 5, 5]]
```

Adição

Matrizes podem ser somadas entre sí desde que tenham a mesma ordem. Se não tiverem, a soma não está determinada.

Se tiverem, o resultado da soma também terá a mesma ordem e no resultado cada elemento corresponderá a à soma dos elementos que ocupam a mesma posição nas duas matrizes sendo somadas. Por exemplo $\begin{pmatrix} 8 & 17 & -3 & 21 \\ 0 & 1 & 2 & 3 \end{pmatrix} + \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix} = \begin{pmatrix} 9 & 19 & 0 & 25 \\ 5 & 7 & 9 & 11 \end{pmatrix}$

Supondo definidas as matrizes A e B em python, eis o trecho que código que acha a soma de $A+B$

```
r=numpy.zeros((len(a),len(a[0])))
for i in range(len(a)):
  for j in range(len(a[0])):
    r[i][j]=a[i][j]+b[i][j]
```

Multiplicação por escalar

Matrizes podem ser multiplicadas por um escalar (um número único) Neste caso, cada elemento da matriz será multiplicado pelo escalar. Por exemplo $\begin{pmatrix} 4 & 3 & 0 \\ -2 & 1 & 5 \end{pmatrix} \times 3 = \begin{pmatrix} 12 & 9 & 0 \\ -6 & 3 & 15 \end{pmatrix}$

Multiplicação de matrizes

A multiplicação de matrizes pode ser a trivial (duas matrizes de mesma ordem, geram uma terceira idem, onde cada elemento é o produto do par de elementos na posição corresponde nas matrizes que estão sendo multiplicadas), mas em geral a multiplicação de matrizes tem a seguinte definição: Sejam A e B matrizes em que o número de colunas de A é igual ao número de linhas de B. Então, o produto de $A.B$ é a matriz com o mesmo número de linhas de A e o mesmo número de colunas de B e cujo elemento da linha i e da coluna k é obtido multiplicando a linha i de A pela coluna k de B e somando o vetor resultante.

$$\begin{pmatrix} a_{11} & \dots & a_{1p} \\ a_{21} & \dots & a_{2p} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mp} \end{pmatrix} . \begin{pmatrix} b_{11} & \dots & b_{1n} \\ b_{21} & \dots & b_{2n} \\ \dots & \dots & \dots \\ b_{p1} & \dots & b_{pn} \end{pmatrix}$$

$$= \begin{pmatrix} c_{11} & \dots & \dots & c_{1n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ c_{m1} & \dots & \dots & c_{mn} \end{pmatrix}$$

onde

$$c_{ik} = a_{i1} \times b_{1k} + a_{i2} \times b_{2k} + \dots + a_{ip} \times b_{pk} =$$

$$\sum_{t=1}^{p} a_{it} \times b_{tk}$$

Em $A.B$ se o número de colunas de A não é igual ao número de linhas de B então o produto $A.B$ não está definido. Eis o bloco Python que faz isso

```
   # a=[[1,2,3],[4,5,6]]
   # b=[[1,1],[2,2],[3,3]]
def mm(a,b):
  import numpy
  la = len(a)
  ca = len(a[0])
  lb = len(b)
  cb = len(b[0])
  if ca != lb:
    print("nao conformaveis...")
  else:
    r=numpy.zeros((la,cb))
    for i in range(0,la):
      for j in range(0,cb):
        s=0
        for k in range(0,ca):
          s=s+a[i][k]*b[k][j]
        r[i][j]=s
  return(r)
```

Leitura de dados externos

A seguir, uma necessidade que vai se apresentar aqui: a leitura de dados em midia magnética ao invés de entrados via teclado. Acompanhe

```
def leia(nome):
  import numpy
  ref = open(nome,"r")
  lin = ref.readline()
  lin = lin.split()
  la = int(lin[0])
  ca = int(lin[1])
  lb = int(lin[2])
  cb = int(lin[3])
  a=numpy.zeros((la,ca))
  b=numpy.zeros((lb,cb))
  r=numpy.zeros((la,cb))
  lin = ref.readline()
  i=0
  while i<la:
    val = lin.split()
    j=0
    while j<ca:
      a[i][j]=int(val[j])
      j=j+1
    i=i+1
    lin = ref.readline()
  i=0
  while i<lb:
    val = lin.split()
    j=0
    while j<cb:
      b[i][j]=int(val[j])
      j=j+1
    i=i+1
    lin = ref.readline()
  for i in range(la):
    for j in range(cb):
      s=0
      for k in range(ca):
        s=s+a[i][k]*b[k][j]
      r[i][j]=s
  return(r)
  ref.close()
```

Para saber se um número é primo

```
import math
def primo(x):
  if x%2==0 and x!=2:
    return False
  lim=math.ceil(x**0.5)
  for j in range (3,lim,2):
    if x%j==0:
      return False
  return True
```

Alguns exemplos

No portal AVA existem alguns arquivos de exemplo para testar seus programas. Os arquivos `T17EXEM?.myd` tem como respostas os seguintes valores:

EXEM1	$\sum(A)$	9170
	$\sum(B)$	8137
	$\sum$8,7; 14,9 e 7,9	71319
	primos	45
EXEM2	$\sum(A)$	7136
	$\sum(B)$	6839
	$\sum$12,16; 9,12 e 10,11	79434
	primos	20
EXEM3	$\sum(A)$	11521
	$\sum(B)$	10206
	$\sum$15,7; 11,7 e 13,11	117353
	primos	22

☞ Para você fazer

Você deve ler o arquivo de nome T17F001.myd que está publicado no portal AVA. Depois de ler e criar as matrizes A e B que estão dentro desse arquivo, você deve calcular 3 coisas:

1. A soma de todos os dados da matriz A
2. A soma de todos os dados da matriz B
3. A matriz R obtida pela multiplicação matricial de A por B. Depois de obtida esta matriz, ache 3 parcelas:

   Primeira parcela O elemento de R na linha e coluna: 8,13

   Segunda parcela O elemento de R na linha e coluna: 9,10

   Terceira parcela O elemento de R na linha e coluna: 14,6

   Agora some essas 3 parcelas e informe no local adequado o resultado achado.

4. Na matriz R quantos elementos são primos ?

Note que a primeira linha do arquivo acima contém apenas 4 números, que indicam a ordem de A e a ordem de B nessa sequência. São números inteiros sem nenhum tipo de delimitador entre eles (apenas um ou mais espaços em branco).
No arquivo acima citado aparecem:

```
17 14 14 17
```

Note também que – se estiver programando em Python – os índices neste ambiente sempre começam em 0 enquanto as contagens acima começam em 1. Então se a folha pede para somar o elemento 11,18 da matriz, em Python isto deve ser escrito como 10,17.

$\sum(A)$	$\sum(B)$	$\sum$3 parcelas:	primos ?