SISTEMAS DE INFORMAÇÃO

Lista de Exercícios 10 - (Python)

Prof. Dr. Marco Antonio Leonel Caetano

Exercícios para resolver usando array bi-dimensional (matrizes) no Python e sistemas lineares com a biblioteca ***linalg***, quando for o caso.

(1) Programar em python para somar as seguintes matrizes usando numpy:

$$S = \begin{pmatrix} 3 & 2 & 1 \\ 4 & 5 & 6 \end{pmatrix} + \begin{pmatrix} 2 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

```
import numpy as np

A=np.array([[3,2,1], [4,5,6]])
B=np.array([[2,2,2], [1,2,3]])
S=A+B
print('S= ')
print(S)
```

Solução:

```
S=
[[5 4 3]
 [5 7 9]]
```

(2) Programar em python para fazer o seguinte produto matricial:

$$P = \begin{pmatrix} 1 & 2 \\ 4 & 5 \\ 3 & 6 \end{pmatrix} \times \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix}$$

```
import numpy as np

A=np.array([[1,2],[4,5],[3,6]])
B=np.array([[3,4],[1,2]])
P=np.matmul(A,B)
print('P= ')
print(P)
```

Solução:

```
P=
[[ 5  8]
 [17 26]
 [15 24]]
```

(3) Fazer uma programação em python para fazer o seguinte produto matricial:

$$P = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & 1 \\ 3 & 4 & 2 \end{pmatrix} \times \begin{pmatrix} 2 & 1 & 4 \\ 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$$

```
import numpy as np

A=np.array([[1,2,1],[0,2,1],[3,4,2]])
B=np.array([[2,1,4],[1,2,3],[2,1,3]])
P=np.matmul(A,B)
print('P= ')
print(P)
```

Solução:

```
P=
[[ 6  6 13]
 [ 4  5  9]
 [14 13 30]]
```

(4) Programar em python para encontrar o determinante da matriz:

$$A = \begin{pmatrix} 2 & 5 & 4 \\ 3 & 1 & 2 \\ 5 & 4 & 6 \end{pmatrix}$$

```
import numpy as np
from numpy.linalg import det

A=np.array([[2,5,4],[3,1,2],[5,4,6]])
deter=det(A)
print('determinante = %4.2f ' % deter)
```

Solução:

```
determinante = -16.00
```

(5) Programar em python para encontrar o ângulo θ entre dois vetores :

$x = \begin{pmatrix} 2 \\ -5 \\ 4 \end{pmatrix}$ e $y = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$ cuja fórmula é:

$$cos\,\theta = \frac{(2 \quad -5 \quad 4) \times \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}}{\sqrt{2^2 + (-5)^2 + 4^2} \times \sqrt{1^2 + 2^2 + (-1)^2}}$$

```python
import numpy as np
import math

x=np.array([2,-5,4])
y=np.array([1,2,-1])

baixo= (np.sqrt(np.sum(x**2))*np.sqrt(np.sum(y**2)))
c_theta=np.matmul(x,y)/baixo

theta=math.degrees(math.acos(c_theta))

print('++++++++++++++++++++++++++++++++++++++++')
print('angulo entre 2 vetores (graus) = %4.2f ' % theta)
```

Solução:

```
++++++++++++++++++++++++++++++++++++++++
angulo entre 2 vetores (graus) = 136.91
```

(6) Observe o seguinte sistema linear:

$$\begin{cases} x_1 + 2x_2 + x_3 = 5 \\ 2x_1 + 2x_2 + x_3 = 6 \\ x_1 + 2x_2 + 3x_3 = 9 \end{cases}$$

(a) Programar em python para achar a solução do sistema usando $x = A^{-1}.B$

```python
import numpy as np
from numpy.linalg import inv

A=np.array([[1,2,1],[2,2,1],[1,2,3]])
B=np.array([5,6,9])
inversa=inv(A)
x=np.matmul(inversa,B)

print('++++++++++++++++++++++++++++')
print('x = ',x[0])
print('y = ',x[1])
print('z = ',x[2])
```

(b) Programar em python usando a função solve da biblioteca linalg.

```python
import numpy as np
from numpy.linalg import solve

A=np.array([[1,2,1],[2,2,1],[1,2,3]])
B=np.array([5,6,9])

x=solve(A,B)

print('++++++++++++++++++++++++++++')
print('x = ',x[0])
print('y = ',x[1])
print('z = ',x[2])
```

(c) Qual a solução encontrada?

```
++++++++++++++++++++++++++++
x =  1.0
y =  1.0
z =  2.0
```

(7) Sistemas lineares são muito importantes para a resolução de problemas de Economia e Administração de empresas para previsão de custos, gastos, otimização de recursos, etc.

Imagine que o governo contratou um consultor para fazer um equacionamento da previdência privada onde

x1 : número de aposentados;
x2 : número de trabalhadores na ativa;
x3 : número de contribuintes;
x4 : número de jovens ainda não atuantes no mercado.

O consultor equacionou 4 políticas de estado diferente para a atuação do governo e no conjunto chegou no seguinte sistema linear:

$$\begin{cases} 4x_2 - 3x_3 - 0{,}5x_4 = 2600 + 2x_1 \\ -x_1 - x_3 + 0{,}7x_4 = 3000 - 3x_2 \\ -4x_1 + 2x_2 + 0{,}8x_4 = 2000 + 2x_3 \\ -5x_1 + x_2 + 3x_3 + x_4 = 1500 \end{cases}$$

(a)Apresente a programação em python que soluciona o sistema linear usando a função solve da biblioteca *linalg.*

Solução: O primeiro passo é transformar o sistema anterior, deixando somente variáveis do lado esquerdo e números do lado direito. Após as mudanças o novo sistema será:

$$\begin{cases} -2x_1 + 4x_2 - 3x_3 - 0{,}5x_4 = 2600 \\ -x_1 + 3x_2 - x_3 + 0{,}7x_4 = 3000 \\ -4x_1 + 2x_2 - 2x_3 + 0{,}8x_4 = 2000 \\ -5x_1 + x_2 + 3x_3 + x_4 = 1500 \end{cases}$$

```python
import numpy as np
from numpy.linalg import solve

A=np.array([[-2,4,-3,-0.5],[-1,3,-1,0.7],[-4,2,-2,0.8],[-5,1,3,1]])
B=np.array([2600,3000,2000,1500])
x=solve(A,B)

print('++++++++++++++++++++++++++++')
print('x1 = %4.2f' % x[0])
print('x2 = %4.2f' % x[1])
print('x3 = %4.2f' % x[2])
print('x4 = %4.2f' % x[3])
```

(b) Dizer quais são os valores de x que o governo terá como solução desse sistema linear, ou seja, quais os valores de x_1, x_2, x_3 e x_4.

```
++++++++++++++++++++++++++++
x1 = 71.27
x2 = 830.62
x3 = 44.72
x4 = 891.60
```

(8) Observe o seguinte sistema linear:

$$\begin{cases} x_1 + 2x_2 = 2 \\ 3x_1 + 2x_2 = 1 \end{cases}$$

(a) Apresente a programação em python que pergunte ao usuário a ordem “n” por input da matriz A (nxn), pede as matrizes A e B (termos independentes) ao usuário usando comando “for” e input, e soluciona o sistema.

```
import numpy as np
from numpy.linalg import solve

n=int(input('ordem n da matriz = '))
A=np.zeros((n,n))
B=np.zeros(n)

for i in range(n):
    for j in range(n):
        A[i,j]=float(input('A = '))

for i in range(n):
    B[i]=float(input('B = '))

x=solve(A,B)

print('++++++++++++++++++++++++++')
print('x1 = %4.2f' % x[0])
print('x2 = %4.2f' % x[1])
```

(b) Qual o valor numérico da solução para x_1 e x_2 ?

```
A = 1

A = 2

A = 3

A = 2

B = 2

B = 1
++++++++++++++++++++++++++
x1 = -0.50
x2 = 1.25
```

(9) *(adaptado do livro “pesquisa operacional na tomada de decisões”- Lachtermacher)*

Um artesão de imagens sacras produz duas imagens diferentes: a de Cristo (quantidade x_1) e a de Nossa Senhora (quantidade x_2). A imagem de Cristo é vendida por R$40,00 e a de Nossa Senhora por R$50,00. Por problemas de saúde, o artesão só consegue trabalhar exatos 9 dias por mês e se passar um dia inteiro fazendo imagens de Cristo faz uma imagem apenas. Para a imagem de Nossa Senhora o artesão precisa de dois dias inteiros. As imagens são entalhadas em peças de madeira e encaixadas depois. A imagem de Cristo precisa de três peças de madeira e a de Nossa Senhora precisa de quatro peças, e só existem no total 20 peças de madeiras por mês.

(a) Monte o problema matemático.

$$Z = 40x_1 + 50x_2 \qquad \text{FATURAMENTO}$$

$$\begin{cases} x_1 + 2x_2 = 9 & \text{DIAS DE TRABALHO} \\ 3x_1 + 4x_2 = 20 & \text{PEÇAS DE MADEIRA PARA FABRICAÇÃO} \end{cases}$$

(b)Programe em python usando a função solve da biblioteca linalg.

```
import numpy as np
from numpy.linalg import solve

A=np.array([[1,2],[3,4]])
B=np.array([9,20])


x=solve(A,B)

print('++++++++++++++++++++++++++++')
print('x1 = %4.2f' % x[0])
print('x2 = %4.2f' % x[1])

faturamento=40*x[0] + 50*x[1]

print('#### faturamento = %4.2f' % faturamento)
```

(c) Qual a quantidade de peças x_1 e x_2, e qual o faturamento do artesão?

```
++++++++++++++++++++++++++++
x1 = 2.00
x2 = 3.50
#### faturamento = 255.00
```

(10) A empresa de artigos de couro "PELE DE MIMOSA ltda" fabrica dois tipos de produtos: malas(x_1) e mochilas(x_2). A empresa tem 2 departamentos para a fabricação. As malas são vendidas com lucro de $50 por unidade e o lucro por unidade de mochila é de $40. As quantidades de horas necessárias para confeccionar cada produto, assim como o número total de horas disponíveis em cada departamento, são apresentados a seguir:

Departamento	***Capacidade por Depto (horas/dia)***	***Horas necessárias (Mala)***	***Horas necessárias (Mochila)***
1	440	2	2
2	300	6/5	3/2

(a) Montar o modelo matemático do problema.

As variáveis são: x_{mala} *e* x_{moch}

$$Lucro = \$50.x_{mala} + \$40.x_{moch}$$

Modelo

$$\begin{cases} 2x_{mala} + 2x_{moch} = 440 \\ (6/5)x_{mala} + (3/2)x_{moch} = 300 \end{cases}$$

(b)Programe em python usando a função solve da biblioteca linalg.

```
import numpy as np
from numpy.linalg import solve

A=np.array([[2,2],[6/5,3/2]])
B=np.array([440,300])


x=solve(A,B)

print('++++++++++++++++++++++++++++')
print('x-mala = %4.2f' % x[0])
print('x-mochila = %4.2f' % x[1])


lucro=50*x[0] + 40*x[1]

print('#### lucro = %4.2f' % lucro)
```

(c)Qual a quantidade de peças x_1 e x_2, e qual o lucro?

```
++++++++++++++++++++++++++++
x-mala = 100.00
x-mochila = 120.00
#### lucro = 9800.00
```

(11) A Eletrotech Co produz 2 tipos de aparelhos: geradores e alternadores. Ambos requerem horas de montagem e horas de testes nos seus processos de fabricação.

x_1 : horas para fabricar gerador (montagem/ teste)

x_2 : horas para fabricar alternador (montagem/ teste)

Cada gerador precisa de 2 horas de montagem e 1 hora de testes, e pode ser vendido com R$150 de lucro. Cada alternador precisa de 3 horas de montagem e 2 horas de testes, e pode ser vendido com R$250 de lucro. No próximo período de produção a empresa terá a sua disposição 260 horas para montagem e 140 horas para testes.

(a)Formule o problema matematicamente com a equação de horas de montagem e com horas de testes.

$$\begin{cases} 2x_1 + 3x_2 = 260 & \textit{equação para horas de montagem} \\ x_1 + 2x_2 = 140 & \textit{equação para horas de testes} \end{cases}$$

$$Lucro = R\$150.x_1 + R\$250.x_2$$

(b)Programe em python usando a função solve da biblioteca linalg.

```python
import numpy as np
from numpy.linalg import solve

A=np.array([[2,3],[1,2]])
B=np.array([260,140])


x=solve(A,B)

print('++++++++++++++++++++++++++++')
print('x-geradores = %4.2f' % x[0])
print('x-alternadores = %4.2f' % x[1])


lucro=150*x[0] + 250*x[1]

print('#### lucro = %4.2f' % lucro)
```

(c)Qual a quantidade de horas para fabricar geradores e alternadores e quanto de lucro?

```
++++++++++++++++++++++++++++
x-geradores = 100.00
x-alternadores = 20.00
#### lucro = 20000.00
```