Matrizes. Soluções aos exercícios

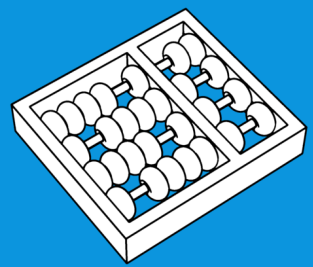
MC102 - Algoritmos e Programação de Computadores

05/25

15

Santiago Valdés Ravelo
https://ic.unicamp.br/~santiago/
ravelo@unicamp.br

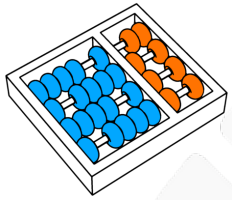
UNICAMP



"Matrizes agem. Elas não apenas ficam lá."

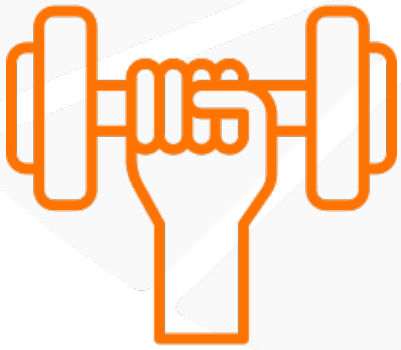
Gilbert Strang.

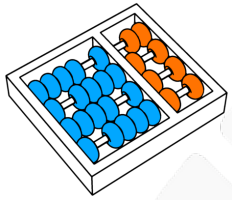
Exercícios



Matrizes

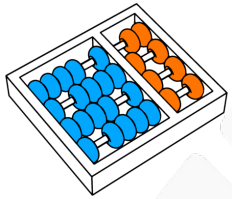
Soluções para os exercícios!





Exercícios de matrizes

1. Faça uma função que dados inteiros **m** e **n**, devolve uma matriz **m**×**n** (isto é, **m** linhas e **n** colunas) com algum valor inicial dado (zero por padrão).
2. Faça uma função que, dado um inteiro **n**, devolve a matriz identidade **n**×**n**.
3. Faça uma função que, dada uma matriz, imprime a matriz (em sua representação usual).
4. Faça uma função que, dada uma matriz M e um escalar λ, calcula $\lambda \cdot M$.
5. Faça uma função que soma duas matrizes.
6. Faça uma função que transpõem uma matriz.
7. Faça uma função que multiplica duas matrizes.



Soluções

Inicialização

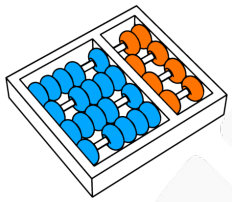
```
1  def inicializa(m, n, valor = 0):
2      matriz = [None] * m
3      for i in range(m):
4          matriz[i] = [valor] * n
5      return matriz
```

Identidade:

```
1  def identidade(n):
2      matriz = inicializa(n, n)
3      for i in range(n):
4          matriz[i][i] = 1
5      return matriz
```

Impressão:

```
1  def imprime(matriz):
2      for linha in matriz:
3          print(*linha)
```



Soluções. Continuação

Produto escalar:

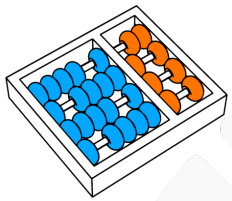
```
1 def produto_escalar(matriz, escalar):
2     resultado = inicializa(len(matriz), len(matriz[0]))
3     for i in range(len(resultado)):
4         for j in range(len(resultado[i])):
5             resultado[i][j] = matriz[i][j] * escalar
6     return resultado
```

Soma:

```
1 def soma(matriz1, matriz2):
2     resultado = inicializa(len(matriz1), len(matriz1[0]))
3     for i in range(len(resultado)):
4         for j in range(len(resultado[i])):
5             resultado[i][j] = matriz1[i][j] + matriz2[i][j]
6     return resultado
```

Transposta:

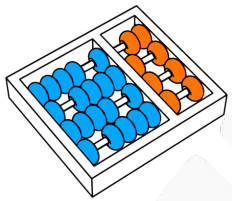
```
1 def transposta(matriz):
2     resultado = inicializa(len(matriz[0]), len(matriz))
3     for i in range(len(resultado)):
4         for j in range(len(resultado[i])):
5             resultado[i][j] = matriz[j][i]
6     return resultado
```



Soluções. Continuação

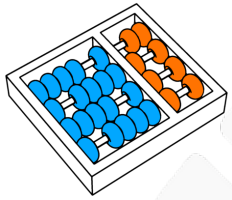
Multiplicação:

```
1  def multiplica(matriz1, matriz2):
2      resultado = inicializa(len(matriz1), len(matriz2[0]))
3      for i in range(len(resultado)):
4          for j in range(len(resultado[i])):
5              for k in range(len(matriz2)):
6                  resultado[i][j] += matriz1[i][k] * matriz2[k][j]
7      return resultado
```



Exercícios de imagens .pbm

1. Faça uma função que lê do terminal o conteúdo de um arquivo **pbm**.
2. Faça uma função que, dada uma matriz, escreve (no terminal) o conteúdo de um arquivo **pbm**.
3. Faça uma função que, dada uma matriz de **0**'s e **1**'s, nega a matriz, isto é, posições que eram **0** viram **1** e vice-versa.
4. Combine os três exercícios anteriores para inverter as cores de uma imagem **pbm**.



Soluções

Leitura:

```
1  def le():
2      c = input()
3      n, m = map(int, input().split())
4      matriz = [None] * m
5      for i in range(m):
6          matriz[i] = list(map(int, input().split()))
7      return matriz
```

Escrita:

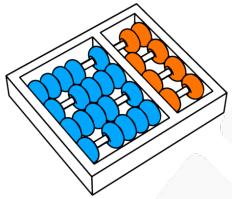
```
1  def escreve(matriz):
2      print('P')
3      print(len(matriz[0]), len(matriz))
4      for linha in matriz:
5          print(*linha)
```

Inverte valores:

```
1  def inverte(matriz):
2      return soma(inicializa(len(matriz), len(matriz[0]), 1), produto_escalar(matriz, -1))
```

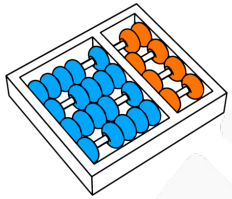
Inverte cores:

```
1  escreve(inverte(le()))
```



Exercícios de linearização

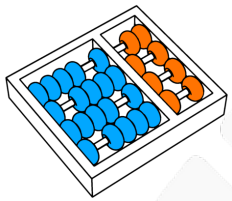
1. Faça um programa que lê duas matrizes, soma as duas e imprime o resultado usando linearização de índices.
2. Linearize uma matriz tridimensional.



Soluções

Leitura, soma e impressão linearizando:

```
1  def le_linear():
2      m, n = map(int, input().split())
3      matriz = [0] * m * n
4      for i in range(m):
5          matriz[i * n : (i + 1) * n - 1] = list(map(int, input().split()))
6      return matriz, m, n
7
8  def soma_linear(matriz1, matriz2):
9      resultado = matriz1[:]
10     for i in range(len(resultado)):
11         resultado[i] += matriz2[i]
12     return resultado
13
14 def imprime_linear(matriz, m, n):
15     for i in range(m):
16         print(matriz[i * n : (i + 1) * n - 1])
17
18 matriz1, m, n = le_linear()
19 matriz2, m, n = le_linear()
20 imprime_linear(soma_linear(matriz, matriz2))
```



Soluções. Continuação

Linearização 3D:

```
1  def lineariza(matriz):
2      m = len(matriz)
3      n = len(matriz[0]
4      o = len(matriz[0][0])
5      matriz_linear = [0] * m * n * o
6      l = 0
7      for i in range(m):
8          for j in range(n):
9              for k in range(o):
10                 matriz_linear[l] = matriz[i][j][k]
11                 l += 1
12     return matriz_linear, m, n, o
13
14 def delineariza(matriz_linear, m, n, o):
15     matriz = [None] * m
16     l = 0
17     for i in range(m):
18         matriz[i] = [None] * n
19         for j in range(n):
20             matriz[i][j] = [0] * o
21             for k in range(o):
22                 matriz[i][j][k] = matriz_linear[l]
23                 l += 1
24     return matriz
```

Matrizes. Soluções aos exercícios

MC102 - Algoritmos e Programação de Computadores

05/25

15

Santiago Valdés Ravelo
https://ic.unicamp.br/~santiago/
ravelo@unicamp.br

UNICAMP