

MAC2166 - Introdução à Computação

Prof. Dr. Helder Oliveira



Agenda

- Processamento da Vizinhaça de um Elemento

Exercício

- Como podemos acessar todos os elementos da primeira linha de uma matriz?

Exercício

- Como podemos acessar todos os elementos da primeira linha de uma matriz?

```
1 linha_1 = matriz[0]
2 print(linha_1) # Saída: [1 2 3]
```

Exercício

- Como acessar todos os elementos da segunda coluna de uma matriz?

Exercício

- Como acessar todos os elementos da segunda coluna de uma matriz?

```
1 # Acessar a segunda coluna (índice 1)
2 for linha in matriz:
3     print(linha[1])
```

Exercício

- Escreva uma função que receba uma matriz e retorne a soma de todos os elementos.

Exercício

- Escreva uma função que receba uma matriz e retorne a soma de todos os elementos.

```
1 def soma_matriz(matriz):
2     resultado = 0
3     for linha in matriz:
4         for coluna in linha:
5             resultado = resultado+coluna
6     return resultado
```

Execício

- Crie uma função para acessar os vizinhos (cima, baixo, esquerda, direita) de um elemento em uma matriz? Retorna uma lista com os elementos vizinhos.

Execício

- Crie uma função para acessar os vizinhos (cima, baixo, esquerda, direita) de um elemento em uma matriz?

```
1 def vizinhanca(matriz, i, j):
2     vizinhos = []
3     if i > 0:
4         vizinhos.append(matriz[i-1][j])    # Cima
5     if i < len(matriz) - 1:
6         vizinhos.append(matriz[i+1][j])    # Baixo
7     if j > 0:
8         vizinhos.append(matriz[i][j-1])    # Esquerda
9     if j < len(matriz[0]) - 1:
10        vizinhos.append(matriz[i][j+1])    # Direita
11    return vizinhos
```

Exercício

- Implemente uma função que receba uma matriz e retorne a soma dos elementos da vizinhança de um dado índice (i, j) .

Exercício

- Implemente uma função que receba uma matriz e retorne a soma dos elementos da vizinhança de um dado índice (i, j).

```
1 def soma_vizinhos(matriz, i, j):  
2     vizinhos = vizinhanca(matriz, i, j)  
3     soma=0  
4     for elemento in vizinhos:  
5         soma = soma+ elemento  
6     return soma
```

Exercício

- Escreva uma função que receba uma matriz e um índice de linha e retorne a soma dos elementos dessa linha.

Exercício

- Escreva uma função que receba uma matriz e um índice de linha e retorne a soma dos elementos dessa linha.

```
1 def soma_linha(matriz, indice_linha):  
2     linha = 0  
3     for elemento in matriz[indice_linha]:  
4         linha = linha + elemento  
5     return linha
```

Exercício

- Escreva uma função que receba uma matriz e um índice de coluna e retorne a soma dos elementos dessa coluna.

Exercício

- Escreva uma função que receba uma matriz e um índice de coluna e retorne a soma dos elementos dessa coluna.

```
1 def soma_coluna(matriz, indice_coluna):  
2     soma = 0  
3     for linha in matriz:  
4         soma += linha[indice_coluna]  
5     return soma
```

Exercício

- Escreva uma função que multiplique todos os elementos de uma matriz por um valor escalar.

Exercício

- Escreva uma função que multiplique todos os elementos de uma matriz por um valor escalar.

```
1 def multiplica_escalar(matriz, escalar):  
2     nova_matriz = []  
3     for linha in matriz:  
4         nova_linha = []  
5         for elemento in linha:  
6             nova_linha.append(elemento * escalar)  
7         nova_matriz.append(nova_linha)  
8     return nova_matriz
```

Exercício

- Escreva uma função que receba uma matriz, um índice de linha e um índice de coluna e retorne os vizinhos do elemento nessa posição. Utilize o **for**.

Exercício

- Escreva uma função que receba uma matriz, um índice de linha e um índice de coluna e retorne os vizinhos do elemento nessa posição. Utilize o **for**.

```
1 def vizinhos(matriz, linha, coluna):
2     vizinhos = []
3     for i in range(max(0, linha - 1), min(len(matriz), linha + 2)):
4         for j in range(max(0, coluna - 1), min(len(matriz[0]), coluna + 2)):
5             if (i, j) != (linha, coluna):
6                 vizinhos.append(matriz[i][j])
7     return vizinhos
```

Exercício

- Escreva uma função que receba uma matriz e retorne a sua transposta (troca de linhas por colunas)

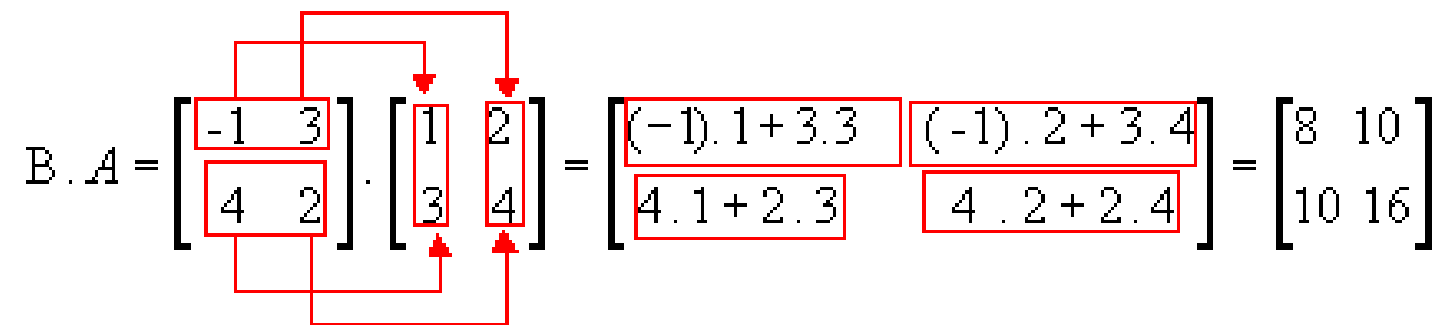
Exercício

- Escreva uma função que receba uma matriz e retorne a sua transposta (troca de linhas por colunas)

```
1 def transposta(matriz):
2     linhas = len(matriz)
3     colunas = len(matriz[0])
4     matriz_t = []
5     for j in range(colunas):
6         nova_linha = []
7         for i in range(linhas):
8             nova_linha.append(matriz[i][j])
9         matriz_t.append(nova_linha)
10    return matriz_t
```

Exercício

- Escreva uma função que receba duas matrizes e retorne o produto delas

$$B \cdot A = \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} (-1) \cdot 1 + 3 \cdot 3 & (-1) \cdot 2 + 3 \cdot 4 \\ 4 \cdot 1 + 2 \cdot 3 & 4 \cdot 2 + 2 \cdot 4 \end{bmatrix} = \begin{bmatrix} 8 & 10 \\ 10 & 16 \end{bmatrix}$$


Exercício

- Escreva uma função que receba duas matrizes e retorne o produto delas

```
1 def multiplica_matrizes(matriz1, matriz2):
2     linhas_m1 = len(matriz1)
3     colunas_m1 = len(matriz1[0])
4     colunas_m2 = len(matriz2[0])
5     resultado = []
6     for i in range(linhas_m1):
7         nova_linha = []
8         for j in range(colunas_m2):
9             soma = 0
10            for k in range(colunas_m1):
11                soma += matriz1[i][k] * matriz2[k][j]
12            nova_linha.append(soma)
13        resultado.append(nova_linha)
14    return resultado
```

Exercício

- Escreva uma função que receba uma matriz quadrada e verifique se ela é simétrica.

Exercício

- Escreva uma função que receba uma matriz quadrada e verifique se ela é simétrica.

```
1 def matriz_simetrica(matriz):
2     linhas = len(matriz)
3     for i in range(linhas):
4         for j in range(linhas):
5             if matriz[i][j] != matriz[j][i]:
6                 return False
7     return True
```

Exercício

- Escreva uma função que receba uma matriz quadrada e verifique se ela é uma matriz identidade.

Exercício

- Escreva uma função que receba uma matriz quadrada e verifique se ela é uma matriz identidade.

```
1 def matriz_identidade(matriz):
2     linhas = len(matriz)
3     for i in range(linhas):
4         for j in range(linhas):
5             if i == j and matriz[i][j] != 1:
6                 return False
7             elif i != j and matriz[i][j] != 0:
8                 return False
9     return True
```

Exercício

- Escreva uma função que receba uma matriz quadrada e rotacione seus elementos em 90 graus no sentido horário.

Exercício

- Escreva uma função que receba uma matriz quadrada e rotacione seus elementos em 90 graus no sentido horário.

```
1 def rotaciona_90_graus(matriz):
2     n = len(matriz)
3     matriz_rotacionada = []
4     for i in range(n):
5         nova_linha = []
6         for j in range(n):
7             nova_linha.append(0)
8         matriz_rotacionada.append(nova_linha)
9
10    for i in range(n):
11        for j in range(n):
12            matriz_rotacionada[j][n - i - 1] = matriz[i][j]
13
14    return matriz_rotacionada
```

Exercício

- Escreva uma função que receba duas matrizes do mesmo tamanho e retorne a soma delas.

Exercício

- Escreva uma função que receba duas matrizes do mesmo tamanho e retorne a soma delas.

```
1 def soma_matrizes(matriz1, matriz2):
2     linhas = len(matriz1)
3     colunas = len(matriz1[0])
4     matriz_soma = []
5     for i in range(linhas):
6         nova_linha = []
7         for j in range(colunas):
8             nova_linha.append(matriz1[i][j] + matriz2[i][j])
9         matriz_soma.append(nova_linha)
10    return matriz_soma
```

Dúvidas

