



Algoritmos e Programação de Computadores

Matrizes

Prof. Edson Borin

Instituto de Computação (IC/Unicamp)

Agenda

- Arranjos
- Matrizes
 - Matrizes e arranjos multidimensionais
 - Criando matrizes
 - Acessando dados de uma matriz
 - Declarando arranjos multidimensionais
- Exemplo com matrizes
- Exercícios

Arranjos

- Um arranjo (*array*) é uma estrutura de dados que armazena uma coleção de elementos de tal forma que cada um dos elementos possa ser identificado por um ou mais índices.
- Casos particulares:
 - Arranjos unidimensionais: vetores
 - Arranjos bidimensionais: matrizes

Matrizes e Arranjos Multidimensionais

- Suponha por exemplo que devemos armazenar as notas de cada aluno(a) (provas, laboratórios, atividades conceituais, exame) de MC102.
- Assumindo que um(a) aluno(a) é avaliado(a) com 4 notas, seria necessário um **vetor** de 4 posições para guardar as notas de um(a) aluno(a).

0	1	2	3
9.5	9	8	0

notas

Matrizes e Arranjos Multidimensionais

- Contudo, assumindo que uma turma tem 130 aluno(a)s, seria necessário uma matriz bidimensional para guardar as notas de todo(a)s o(a)s aluno(a)s de uma turma.

		notas			
		0	1	2	3
aluno(a)s	0	9.5	9	8	0
	1	5	6	7	8.5
	2	9	8	8	0
		...			

notas

	0	1	2	3
0	9.5	9	8	0
1	5	6	7	8.5
2	9	8	8	0

...

```
turma = [[9.5, 9, 8, 0], [5, 6, 7, 8.5], [9, 8, 8, 0],  
         [3.6, 7.0, 9.1, 8.7], [5.0, 4.5, 7.0, 5.2],  
         [2.1, 6.5, 8.0, 7.0], ... ]
```

notas

	0	1	2	3
0	9.5	9	8	0
1	5	6	7	8.5
2	9	8	8	0

...

```
turma = [[9.5, 9, 8, 0], [5, 6, 7, 8.5], [9, 8, 8, 0],  
         [3.6, 7.0, 9.1, 8.7], [5.0, 4.5, 7.0, 5.2],  
         [2.1, 6.5, 8.0, 7.0], ... ]
```

notas

	0	1	2	3
0	9.5	9	8	0
1	5	6	7	8.5
2	9	8	8	0

aluno(a)s

...

```
turma = [[9.5, 9, 8, 0], [5, 6, 7, 8.5], [9, 8, 8, 0],  
         [3.6, 7.0, 9.1, 8.7], [5.0, 4.5, 7.0, 5.2],  
         [2.1, 6.5, 8.0, 7.0], ... ]
```


notas

	0	1	2	3
0	9.5	9	8	0
1	5	6	7	8.5
2	9	8	8	0

...

```
turma = [[9.5, 9, 8, 0], [5, 6, 7, 8.5], [9, 8, 8, 0],  
         [3.6, 7.0, 9.1, 8.7], [5.0, 4.5, 7.0, 5.2],  
         [2.1, 6.5, 8.0, 7.0], ... ]
```

notas

	0	1	2	3
0	9.5	9	8	0
1	5	6	7	8.5
2	9	8	8	0

...

```
turma = [[9.5, 9, 8, 0], [5, 6, 7, 8.5], [9, 8, 8, 0],  
         [3.6, 7.0, 9.1, 8.7], [5.0, 4.5, 7.0, 5.2],  
         [2.1, 6.5, 8.0, 7.0], ... ]
```

```
type(turma)
```

```
list
```

Acesso aos Valores: [linha][coluna]

- Segunda nota do primeiro aluno(a): `turma[0][1]`
- Quarta nota do terceiro aluno(a): `turma[2][3]`

		notas			
		0	1	2	3
aluno(a)s	0	9.5	9	8	0
	1	5	6	7	8.5
	2	9	8	8	0
		...			

Declarando uma Matriz com Listas

- Para criar uma matriz de dimensões $l \times c$ inicialmente vazia podemos utilizar listas.
- Exemplo de uma matriz 3 x 4 inicialmente vazia:

```
mat = [[] for i in range(3)]  
#dentro da lista externa cria-se vazia 3 listas []  
mat  
[[], [], []]
```

- Note que cada lista interna representa uma linha da matriz, e seu tamanho pode ser 4 ou qualquer outro valor.

Exemplo de Declaração de Matriz

- Criar matriz 3 x 4 onde cada posição (i, j) contém o valor de $i * j$.

	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	4	6

Exemplo de Declaração de Matriz

- Criar matriz 3 x 4 onde cada posição (i, j) contém o valor de $i * j$.

```
mat = []
for i in range(3): # para cada linha de 0 ate 2
    l = [] # linha começa vazia
    for j in range(4): # para cada coluna de 0 ate 3
        l.append(i*j) # preenche colunas da linha i
    mat.append(l) # adiciona linha na matriz
print(mat)
```

```
[[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]
```

Exemplo de Declaração de Matriz

- Obtendo o mesmo resultado utilizando compreensão de listas:

```
mat = [[i*j for j in range(4)] for i in range(3)]  
print(mat)
```

```
[[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]
```

Acessando os Dados da Matriz

```
nome_da_matriz[linha][coluna]
```

- Ex: `matriz[1][10]`: refere-se à variável na 2ª linha e na 11ª coluna da matriz.
- Lembre-se que, como a matriz está implementada com listas, a primeira posição em uma determinada dimensão começa no índice 0.
- O acesso a posições inválidas causa um erro de execução.

Declarando Arranjos Multidimensionais

- Ainda para o exemplo da turma, assumindo que um curso tem duas turmas, seria necessário um objeto tridimensional para guardar as notas de todo(a)s o(a)s aluno(a)s de todas as turmas do curso.

Declarando Arranjos Multidimensionais

		notas			
		0	1	2	3
aluno(a)s	0	9.5	9	8	0
	1	2	8	10	5
	2	8.6	4.3	7	8.5
	8	3	5.7	7.9	

		notas			
		0	1	2	3
aluno(a)s	0	9.5	9	8	0
	1	2	8	10	5
	2	8.6	4.3	7	8.5
		8	3	5.7	7.9

```
turmas = [[ [9.5, 9, 8, 0], [5, 6, 7, 8.5], [9, 8, 8, 0] ],
           [ [2, 8, 10, 5], [8.6, 4.3, 7, 8.5], [8, 3, 5.7, 7.9] ] ]
```

		notas			
		0	1	2	3
aluno(a)s	0	9.5	9	8	0
	1	2	8	10	5
	2	8.6	4.3	7	8.5
		8	3	5.7	7.9

```
turmas = [[ [9.5, 9, 8, 0], [5, 6, 7, 8.5], [9, 8, 8, 0] ],
            [ [2, 8, 10, 5], [8.6, 4.3, 7, 8.5], [8, 3, 5.7, 7.9] ] ]
```

turma

		notas			
		0	1	2	3
aluno(a)s	0	9.5	9	8	0
	1	2	8	10	5
	2	8.6	4.3	7	8.5
		8	3	5.7	7.9

```
turmas = [[ [9.5, 9, 8, 0], [5, 6, 7, 8.5], [9, 8, 8, 0]],
           [ [2, 8, 10, 5], [8.6, 4.3, 7, 8.5], [8, 3, 5.7, 7.9] ] ]
```

aluno(a)

		notas			
		0	1	2	3
aluno(a)s	0	9.5	9	8	0
	1	2	8	10	5
	2	8.6	4.3	7	8.5
		8	3	5.7	7.9

```
turmas = [[9.5, 9, 8, 0], [5, 6, 7, 8.5], [9, 8, 8, 0]],
          [[2, 8, 10, 5], [8.6, 4.3, 7, 8.5], [8, 3, 5.7, 7.9]]]
print(turmas[1][0][3])
```

5

Exemplos com Matrizes

- Criar programas com operações básicas sobre matrizes quadradas
 - Soma de 2 matrizes com dimensões $n \times n$.
 - Subtração de 2 matrizes com dimensões $n \times n$.

Exemplos com Matrizes

- Primeiramente vamos implementar o código para fazer a leitura e a impressão de uma matriz.

```
def leMatriz(dimensao):  
    mat = [[] for i in range(dimensao)]  
    for i in range(dimensao):  
        for j in range(dimensao):  
            num = int(input("(" + str(i+1) + ", " + str(j+1) + "): "))  
            mat[i].append(num)  
    return mat
```


Exemplos com Matrizes

- Primeiramente vamos implementar o código para fazer a leitura e a impressão de uma matriz.

```
def imprimeMatriz(mat):  
    for linha in mat:  
        for numero in linha:  
            print(numero, end=" ")  
            #imprime números na mesma linha separados por espaço  
        print() #apos impressão de uma linha, pula uma linha
```

Exemplos com Matrizes

- Vamos implementar o código para fazer a **soma** de duas matrizes.
- Para cada posição (i, j) fazemos:

$$\text{mat3}[i][j] = \text{mat1}[i][j] + \text{mat2}[i][j]$$

tal que o resultado da soma das matrizes estará em `mat3`.

```
def somaMatriz(mat1, mat2):  
    tam = len(mat1)  
    mat3 = [[0 for j in range(tam)] for i in range(tam)]  
    for i in range(tam):  
        for j in range(tam):  
            mat3[i][j] = mat1[i][j] + mat2[i][j]  
    return mat3
```

Exemplos com Matrizes

- Vamos implementar o código para fazer a **multiplicação** de duas matrizes (de dimensão $n \times n$).
- Uma posição (i, j) de `mat3` terá o produto interno do vetor linha i de `mat1` com o vetor coluna j de `mat2`:

$$\text{mat3}[i][j] = \sum_{k=0}^{n-1} \text{mat1}[i][k] * \text{mat2}[k][j]$$

0	1
2	3

 ×

4	5
6	7

 =

Exemplos com Matrizes

- Vamos implementar o código para fazer a **multiplicação** de duas matrizes (de dimensão $n \times n$).
- Uma posição (i, j) de `mat3` terá o produto interno do vetor linha i de `mat1` com o vetor coluna j de `mat2`:

$$\text{mat3}[i][j] = \sum_{k=0}^{n-1} \text{mat1}[i][k] * \text{mat2}[k][j]$$

0	1
2	3

 ×

4	5
6	7

 =

6	

$0*4 + 1*6 = 6$

Exemplos com Matrizes

- Vamos implementar o código para fazer a **multiplicação** de duas matrizes (de dimensão $n \times n$).
- Uma posição (i, j) de `mat3` terá o produto interno do vetor linha i de `mat1` com o vetor coluna j de `mat2`:

$$\text{mat3}[i][j] = \sum_{k=0}^{n-1} \text{mat1}[i][k] * \text{mat2}[k][j]$$

0	1
2	3

 ×

4	5
6	7

 =

6	7

$0*5 + 1*7 = 7$



Algoritmos e Programação de Computadores

Matrizes

Prof. Edson Borin

Instituto de Computação (IC/Unicamp)

Exemplos com Matrizes

- Vamos implementar o código para fazer a **multiplicação** de duas matrizes (de dimensão $n \times n$).
- Uma posição (i, j) de `mat3` terá o produto interno do vetor linha i de `mat1` com o vetor coluna j de `mat2`:

$$\text{mat3}[i][j] = \sum_{k=0}^{n-1} \text{mat1}[i][k] * \text{mat2}[k][j]$$

0	1
2	3

 \times

4	5
6	7

 $=$

6	7
26	

$2*4 + 3*6 = 26$

Exemplos com Matrizes

- Vamos implementar o código para fazer a **multiplicação** de duas matrizes (de dimensão $n \times n$).

```
def multiplicaMatriz(mat1, mat2):  
    tam = len(mat1)  
    mat3 = [[0 for j in range(tam)] for i in range(tam)]  
    for i in range(tam):  
        for j in range(tam):  
            for k in range(tam):  
                mat3[i][j] = mat3[i][j] + (mat1[i][k] * mat2[k][j])  
    return mat3
```


Exemplos com Matrizes: O Programa

```
n = int(input("Dimensão das matrizes: "))
mat1 = leMatriz(n)
mat2 = leMatriz(n)
print("mat1: ")
imprimeMatriz(mat1)
print("mat2: ")
imprimeMatriz(mat2)
mat3Soma = somaMatriz(mat1, mat2)
print("Soma: ")
imprimeMatriz(mat3Soma)
mat3Mult = multiplicaMatriz(mat1, mat2)
print("Multiplicação: ")
imprimeMatriz(mat3Mult)
```

Copiando Matrizes

- Uma matriz é uma lista de listas.
- Copiar o conteúdo de uma variável para outra faz com que ambas apontem para a mesma lista de listas.

```
A = [[1,2], [3,4]]
```

```
B = A
```

```
B[0][0] = 0
```

```
print("A:",A)
```

```
print("B:",B)
```

```
A: [[0,2], [3,4]]
```

```
B: [[0,2], [3,4]]
```

Copiando Matrizes

- Uma matriz é uma lista de listas.
- Duplicar apenas a lista externa (a lista de listas) também não é o suficiente - as listas que representam as linhas continuam sendo compartilhadas!

```
A = [[1,2], [3,4]]  
B = A.copy()  
B[0][0] = 0  
print("A:",A)  
print("B:",B)
```

```
A: [[0,2], [3,4]]  
B: [[0,2], [3,4]]
```

Copiando Matrizes

- Uma matriz é uma lista de listas.
- Duplicar apenas a lista externa (a lista de listas) também não é o suficiente - as listas que representam as linhas continuam sendo compartilhadas!

```
A = [[1,2], [3,4]]  
B = A.copy()  
B[0][0] = 0  
print("A:",A)  
print("B:",B)
```

B e A apontam para listas diferentes, entretanto, ambas apontam para as mesmas listas que representam as linhas!

Copiando Matrizes

- Uma matriz é uma lista de listas.
- Duplicar apenas a lista externa (a lista de listas) também não é o suficiente.

```
A = [[1,2], [3,4]]
B = A.copy()
for i in range(len(B)):
    B[i] = A[i].copy()
B[0][0] = 0
print("A:",A)
print("B:",B)
```

```
A: [[1,2], [3,4]]
B: [[0,2], [3,4]]
```

Copiando Matrizes

- Uma matriz é uma lista de listas.
- Duplicar apenas a lista externa (a lista de listas) também não é o suficiente.

```
A = [[1,2], [3,4]]
B = [linha.copy() for linha in A]
B[0][0] = 0
print("A:",A)
print("B:",B)
```

```
A: [[1,2], [3,4]]
B: [[0,2], [3,4]]
```

Exercícios

Exercício 1

- Escreva uma função que toma como parâmetro uma lista bidimensional (lista de listas) e verifica se ela é uma matriz.
 - Em caso positivo, sua função deve retornar um tupla com o número de linhas e de colunas da matriz.
 - Em caso negativo, deve retornar o objeto None

Exercício 2

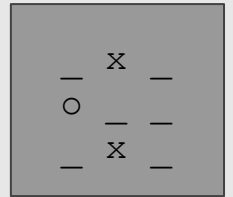
- Escreva uma função que toma como parâmetro uma matriz (representada como uma lista de listas) e retorna sua transposta (também representada como uma lista de listas).

Exercício 3: Jogo da Velha

- Faça um programa para o Jogo da Velha. Seu programa deve:
 - Inicialmente, criar uma matriz 3x3 iniciada com o valor "_".
 - Em seguida, o programa deve ler as jogadas de dois jogadores (Jogador "x" e Jogador "o") alternadamente, até que alguém vença ou todas as casas do tabuleiro estejam ocupadas.

Exercício 3: Jogo da Velha

- A cada jogada:
 - O programa deve mostrar o tabuleiro desta forma =>
 - Solicitar ao jogador "x" ou "o" a linha e a coluna;
 - Atualizar a matriz;
 - Verificar se algum dos jogadores ganhou; caso positivo, informar quem ganhou e terminar o programa; caso negativo, ir para a próxima rodada.



Exercício 4

- Escreva uma função que leia e retorne uma matriz de inteiros fornecida pelo usuário. Sua matriz deve ler os números linha a linha. Os números devem estar separados por espaços em branco. Sua função deve interromper a leitura ao receber uma linha em branco.

Exercício 5

Escreva uma função que dada uma matriz quadrada, verifique se ela é uma matriz diagonal.

- Exemplo:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

Exercício 6

Escreva uma função que dada uma matriz quadrada, verifique se ela é uma matriz triangular inferior.

- Exemplo:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 4 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 9 & 0 & 2 & 3 \end{bmatrix}$$

Exercício 7

Uma matriz de permutações é uma matriz quadrada cujos elementos são zeros ou uns, tal que em cada linha e em cada coluna exista exatamente um elemento igual a 1.

Escreva uma função que, dada uma matriz quadrada, verifique se ela é uma matriz de permutações. Retorne True se ela for, do contrário, retorne False!

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Exercício 8

Uma matriz quadrada de números inteiros é um quadrado mágico se o valor da soma dos elementos de cada linha, de cada coluna e da diagonal principal e da diagonal secundária é o mesmo. Além disso, a matriz deve conter todos os números inteiros do intervalo $[1..n \times n]$. A matriz ao lado é um quadrado mágico, cujas somas valem 65.

Escreva uma função que, dada uma matriz quadrada, verifique se ela é um quadrado mágico.

15	8	1	24	17
16	14	7	5	23
22	20	13	6	4
3	21	19	12	10
9	2	25	18	11

Exercício 9

Escreva uma função que recebe duas matrizes (A e B). Se as duas matrizes tiverem dimensões compatíveis, sua função deve retornar o produto das duas ($C = A \times B$). Caso contrário, sua função deve retornar uma lista vazia.

- Exemplo:

$$\begin{array}{c} A \\ \left[\begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right] \end{array} \times \begin{array}{c} B \\ \left[\begin{array}{c} 5 \\ 6 \end{array} \right] = \begin{array}{c} C \\ \left[\begin{array}{c} 17 \\ 39 \end{array} \right] \end{array}$$

Exercício 10

Escreva uma função que verifica se uma matriz quadrada, passada como parâmetro, é uma matriz simétrica.

Uma matriz é simétrica se ela for igual à sua transposta.

- Exemplo:

	0	1	2	3
0	11	-3	4	8
1	-3	12	6	11
2	4	6	5	13
3	8	11	13	5

Exercício 11

Calculando a nota final

O professor de MC102 decidiu colocar todas as notas dos(as) alunos(as) da disciplina em uma matriz de $N \times M$, sendo uma linha para cada aluno(a) (# de alunos(as) = N) e uma coluna para a nota de cada avaliação (M avaliações).

Agora ele precisa calcular a nota final de cada aluno(a). Você deve ajudar o professor escrevendo uma função que calcula a nota final dos alunos e gera uma lista com estas notas, ou seja, uma lista com N notas (uma para cada aluno).

Exercício 11

Calculando a nota final

Sua função deve:

- Ser chamada de `calcula_nota_final`;
- Receber a matriz de notas, representada como uma lista de listas, e uma lista de pesos, que corresponde aos pesos das avaliações.
- Calcular a nota final de cada aluno(a) realizando uma média ponderada das notas com os pesos fornecidos;
- Retornar uma lista com as notas calculadas.

Exercício 11

Calculando a nota final

- Seja:
 - N_i a i -ésima nota do(a) aluno(a); e
 - P_i o i -ésimo peso da lista

- Nota do(a) aluno(a) $i = \frac{(\sum_{i=0}^{N-1} N_i \times P_i)}{\sum_{i=0}^{N-1} P_i}$

Referências & Exercícios

- Os *slides* deste curso foram baseados nos slides produzidos e cedidos gentilmente pela Professora Sandra Ávila, do Instituto de Computação da Unicamp. Parte dos slides foram baseados no material de MC102 do Prof. Eduardo Xavier (IC/Unicamp)
- <https://panda.ime.usp.br/aulasPython/static/aulasPython/aula11.html>
- <http://www.galirows.com.br/meublog/programacao/exercicios-resolvidos-python/>
(Exercícios resolvidos, explicação com vídeo)