Matrizes

Aula 12

Diego Padilha Rubert

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Algoritmos e Programação

Conteúdo da aula

Introdução

- listas são sequências unidimensionais
- algumas vezes, precisamos representar sequências bidimensionais, ou seja, **matrizes**
- não há uma forma direta de fazer isso em Python

Introdução

- listas são sequências unidimensionais
- algumas vezes, precisamos representar sequências bidimensionais, ou seja, **matrizes**
- não há uma forma direta de fazer isso em Python

Introdução

- listas são sequências unidimensionais
- algumas vezes, precisamos representar sequências bidimensionais, ou seja, **matrizes**
- não há uma forma direta de fazer isso em Python

Definição, declaração e uso

- na matemática, uma **matriz** é uma tabela ou um quadro contendo m linhas e n colunas e usada, entre outros usos, para a resolução de sistemas de equações lineares e transformações lineares
- uma matriz com m linhas e n colunas é chamada de uma **matriz m por n** e denota-se $m \times n$
- os valores m e n são chamados de **dimensões**, **tipo** ou **ordem** da matriz

Definição, declaração e uso

- na matemática, uma **matriz** é uma tabela ou um quadro contendo m linhas e n colunas e usada, entre outros usos, para a resolução de sistemas de equações lineares e transformações lineares
- uma matriz com m linhas e n colunas é chamada de uma **matriz m por n** e denota-se $m \times n$
- os valores m e n são chamados de **dimensões**, **tipo** ou **ordem** da matriz

Definição, declaração e uso

- na matemática, uma **matriz** é uma tabela ou um quadro contendo m linhas e n colunas e usada, entre outros usos, para a resolução de sistemas de equações lineares e transformações lineares
- uma matriz com m linhas e n colunas é chamada de uma **matriz m por n** e denota-se $m \times n$
- os valores m e n são chamados de **dimensões**, **tipo** ou **ordem** da matriz

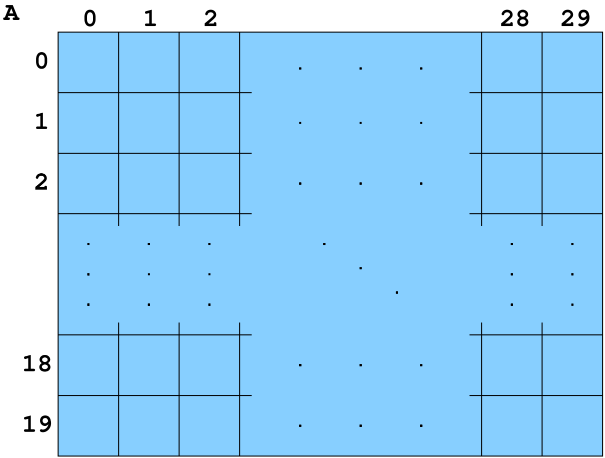
Definição, declaração e uso

- um elemento de uma matriz A que está na linha i e na coluna j é chamado de **elemento** i, j ou **(i, j)-ésimo** elemento de A
- este elemento é denotado por $A_{i,j}$ ou $A(i, j)$

Definição, declaração e uso

- um elemento de uma matriz A que está na linha i e na coluna j é chamado de **elemento** i, j ou **(i, j)-ésimo** elemento de A
- este elemento é denotado por $A_{i,j}$ ou $A(i, j)$

- Exemplo de uma matriz A com 20 linhas e 30 colunas:



Definição, declaração e uso

- na linguagem Python, não há uma forma direta de criar uma matriz
- podemos simular uma matriz criando uma lista de listas

```
A = [[1, 2, 3], [4, 5, 6]]
```

- a primeira linha de uma matriz tem índice 0, a segunda linha tem índice 1, e assim por diante
- do mesmo modo, a primeira coluna da matriz tem índice 0, a segunda tem índice 1 e assim por diante
- para referenciar o valor da célula da linha 0 e da coluna 2 da matriz `A`, devemos usar o identificador da variável e os índices 0 e 2 envolvidos por colchetes, ou seja, `A[0][2]`

Definição, declaração e uso

- na linguagem Python, não há uma forma direta de criar uma matriz
- podemos simular uma matriz criando uma lista de listas

```
A = [[1, 2, 3], [4, 5, 6]]
```

- a primeira linha de uma matriz tem índice 0, a segunda linha tem índice 1, e assim por diante
- do mesmo modo, a primeira coluna da matriz tem índice 0, a segunda tem índice 1 e assim por diante
- para referenciar o valor da célula da linha 0 e da coluna 2 da matriz `A`, devemos usar o identificador da variável e os índices 0 e 2 envolvidos por colchetes, ou seja, `A[0][2]`

Definição, declaração e uso

- na linguagem Python, não há uma forma direta de criar uma matriz
- podemos simular uma matriz criando uma lista de listas

```
A = [[1, 2, 3], [4, 5, 6]]
```

- a primeira linha de uma matriz tem índice 0, a segunda linha tem índice 1, e assim por diante
- do mesmo modo, a primeira coluna da matriz tem índice 0, a segunda tem índice 1 e assim por diante
- para referenciar o valor da célula da linha 0 e da coluna 2 da matriz `A`, devemos usar o identificador da variável e os índices 0 e 2 envolvidos por colchetes, ou seja, `A[0][2]`

Definição, declaração e uso

- na linguagem Python, não há uma forma direta de criar uma matriz
- podemos simular uma matriz criando uma lista de listas

```
A = [[1, 2, 3], [4, 5, 6]]
```

- a primeira linha de uma matriz tem índice 0, a segunda linha tem índice 1, e assim por diante
- do mesmo modo, a primeira coluna da matriz tem índice 0, a segunda tem índice 1 e assim por diante
- para referenciar o valor da célula da linha 0 e da coluna 2 da matriz `A`, devemos usar o identificador da variável e os índices 0 e 2 envolvidos por colchetes, ou seja, `A[0][2]`

Definição, declaração e uso

- na linguagem Python, não há uma forma direta de criar uma matriz
- podemos simular uma matriz criando uma lista de listas

```
A = [[1, 2, 3], [4, 5, 6]]
```

- a primeira linha de uma matriz tem índice 0, a segunda linha tem índice 1, e assim por diante
- do mesmo modo, a primeira coluna da matriz tem índice 0, a segunda tem índice 1 e assim por diante
- para referenciar o valor da célula da linha 0 e da coluna 2 da matriz `A`, devemos usar o identificador da variável e os índices 0 e 2 envolvidos por colchetes, ou seja, `A[0][2]`

Criando uma matriz com dimensões variáveis

Como já vimos, podemos criar uma matriz com valores predeterminados:

```
A = [[1, 2, 3], [4, 5, 6]]
```

Contudo, muitas vezes é o usuário que deseja dizer as dimensões e quais são os elementos da matriz

Criando uma matriz com dimensões variáveis

Podemos criar uma matriz com valores iniciais (0, por exemplo) de acordo com dimensões passadas pelo usuário:

```
m = int(input())
n = int(input())
A = []
for i in range(m):
   A.append([0] * n)
```

Criando uma matriz com dimensões variáveis

Podemos também criar uma matriz já preenchida com valores passados pelo usuário:

```
m = int(input())
n = int(input()) # algumas vezes não é necessário...
A = []
for i in range(m):
   A.append([int(j) for j in input().split()])
```

A última linha corresponde ao código que já vimos nas aulas anteriores, que lê vários valores inteiros em uma linha e cria uma lista com esses valores.

Iterando sobre uma matriz

Podemos iterar sobre os elementos da matriz variando uma variável i sobre os índices das linhas e, para cada linha, variar outra variável j sobre os índices das colunas:

```
m = int(input())
n = int(input())
A = []
for i in range(m):
   A.append([int(j) for j in input().split()])

for i in range(m):
   for j in range(n):
      print("A[%d][%d] = %d" % (i, j, A[i][j]))
```

Iterando sobre uma matriz

Podemos também iterar uma variável $linha$ sobre as próprias linhas e outra variável v sobre os valores de cada linha:

```
m = int(input())
n = int(input())
A = []
for i in range(m):
   A.append([int(j) for j in input().split()])

for linha in A:
   for v in linha:
      print("valor = %d" % (v))
```

Contudo, desta forma não temos como saber qual é a posição de cada elemento que estamos analisando. Portanto, apenas quando essa informação não é necessária que podemos utilizar esta forma de iterar sobre os valores da matriz.

Exemplo

```
m = int(input())
n = int(input())
A = []
for i in range(m):
   A.append([int(j) for j in input().split()])

k = int(input()) # a linha que queremos somar
l = int(input()) # a coluna que queremos somar

soma = 0
for v in A[k]:
   soma += v
print("Valor da soma da linha %d: %d" % (k, soma))

soma = 0
for i in range(m):
   soma += A[i][l]
print("Valor da soma da coluna %d: %d" % (l, soma))

exit(0)
```

Exercícios

1. Dada uma matriz de números inteiros $A_{m \times n}$, imprimir o número de linhas e o número de colunas nulas da matriz.
   Exemplo:
   Se a matriz A tem $m = 4$ linhas, $n = 4$ colunas e conteúdo

$$\begin{pmatrix} 1 & 0 & 2 & 3 \\ 4 & 0 & 5 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

   então A tem 2 linhas nulas e 1 coluna nula.

Exercícios

2. Dizemos que uma matriz de números inteiros $A_{n \times n}$ é uma **matriz de permutação** se em cada linha e em cada coluna houver $n-1$ elementos nulos e um único elemento 1.
Exemplo:
A matriz $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ é de permutação, mas a matriz
$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ não é de permutação.
Dada uma matriz de números inteiros $A_{n \times n}$, verificar se A é de permutação.

3. Dizemos que uma matriz quadrada de números inteiros distintos é um **quadrado mágico** se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos da diagonal principal e secundária são todas iguais.
   Exemplo:
   A matriz
   $$\begin{pmatrix} 8 & 0 & 7 \\ 4 & 5 & 6 \\ 3 & 10 & 2 \end{pmatrix}$$
   é um quadrado mágico.
   Dada uma matriz quadrada de números inteiros $A_{n \times n}$, verificar se A é um quadrado mágico.

Exercícios

4. (a) Imprimir as n primeiras linhas do triângulo de Pascal.

$$\begin{array}{llllll}
1 \\
1 & 1 \\
1 & 2 & 1 \\
1 & 3 & 3 & 1 \\
1 & 4 & 6 & 4 & 1 \\
1 & 5 & 10 & 10 & 5 & 1 \\
\vdots
\end{array}$$

   (b) Imprimir as n primeiras linhas do triângulo de Pascal usando apenas uma lista.

5. Dada uma matriz de números inteiros $A_{m \times n}$, verificar se existem elementos repetidos em A.

6. Faça um programa que, dada uma matriz de números reais $A_{m \times n}$, determine A^t.
7. Dada uma matriz de números reais A com m linhas e n colunas, e uma lista de números reais v com n elementos, determinar o produto de A por v.
8. Dadas duas matrizes de números reais $A_{m \times n}$ e $B_{n \times p}$ calcular o produto de A por B.

Exercícios

9. Dizemos que uma matriz $A_{n \times n}$ é um **quadrado latino de ordem n** se em cada linha e em cada coluna aparecem todos os inteiros $1, 2, 3, \ldots, n$, ou seja, cada linha e coluna é permutação dos inteiros $1, 2, \ldots, n$.
   Exemplo:
   $$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 2 \end{pmatrix}$$
   A matriz acima é um quadrado latino de ordem 4.

Exercícios

9. (*continuação*)
   (a) Escreva uma função com a seguinte interface:

```
def linha(A, i):
```

que receba como parâmetros uma matriz $A_{n \times n}$ de números inteiros e um índice i. Seja n a quantidade de elementos na linha A_i. A função deve verificar se na linha i de A ocorrem todos os números inteiros de 1 a n, devolvendo `True` em caso positivo e `False` caso contrário.

Exercícios

9. (*continuação*)
   (b) Escreva uma função com a seguinte interface:

```
def coluna(A, j):
```

   que receba como parâmetros uma matriz $A_{n \times n}$ de números inteiros e um índice j. Seja n a quantidade de linhas da matriz A. A função deve verificar se na coluna j de A ocorrem todos os números inteiros de 1 a n, devolvendo `True` em caso positivo e `False` caso contrário.

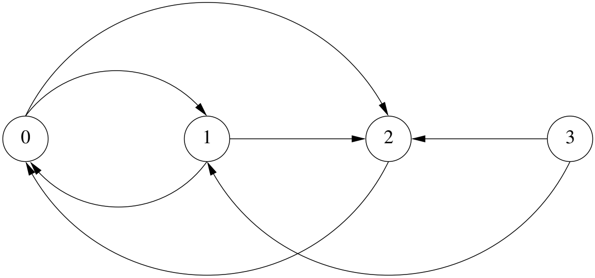
   (c) Usando as funções dos itens (a) e (b), escreva um programa que verifique se uma dada matriz $A_{n \times n}$ de números inteiros, com $1 \leqslant n \leqslant 100$, é um quadrado latino de ordem n.

Exercícios

10. Considere n cidades numeradas de 0 a $n - 1$ que estão interligadas por uma série de estradas de mão única, com $1 \leqslant n \leqslant 100$. As ligações entre as cidades são representadas pelos elementos de uma matriz quadrada $A_{n \times n}$, cujos elementos a_{ij} assumem o valor 1 ou 0, conforme exista ou não estrada direta que saia da cidade i e chegue à cidade j. Assim, os elementos da coluna j indicam as estradas que chegam à cidade j. Por convenção $a_{ii} = 1$.
    A figura a seguir mostra um exemplo para $n = 4$.

Exercícios

10.



A	0	1	2	3
0	1	1	1	0
1	1	1	1	0
2	1	0	1	0
3	0	1	1	1

Exercícios

10. (a) Dado k, determinar quantas estradas saem e quantas chegam à cidade k.
    (b) A qual das cidades chega o maior número de estradas?
    (c) Dado k, verificar se todas as ligações diretas entre a cidade k e outras são de mão dupla.
    (d) Relacionar as cidades que possuem saídas diretas para a cidade k.
    (e) Relacionar, se existirem:
        (i) As cidades isoladas, isto é, as que não têm ligação com nenhuma outra;
        (ii) As cidades das quais não há saída, apesar de haver entrada;
        (iii) As cidades das quais há saída sem haver entrada.
    (f) Dada uma seqüência de m inteiros cujos valores estão entre 0 e $n-1$, verificar se é possível realizar o roteiro correspondente. No exemplo dado, o roteiro representado pela seqüência 2 0 1 2 3, com $m = 5$, é impossível.
    (g) Dados k e p, determinar se é possível ir da cidade k para a cidade p pelas estradas existentes. Você consegue encontrar o menor caminho entre as duas cidades?