

CI1055: Algoritmos e Estruturas de Dados I

Profs. Drs. Marcos Castilho Carmem Hara e Bruno Muller Jr

Departamento de Informática/UFPR

1 de setembro de 2020

Matrizes - Introdução

- ▶ Conceituação
- ▶ Leitura e impressão
- ▶ Encontrar menor elemento
- ▶ Soma de matrizes
- ▶ Multiplicação de matrizes

Conceituação

- ▶ A figura mostra conceitualmente o que é uma matriz.

	1	2	3	4
1	4	6	2	1
2	9	0	0	2
3	8	7	3	9
4	1	3	2	4
5	0	1	0	1

- ▶ Cada quadrado armazena exatamente uma informação de um mesmo tipo (inteiro, real, etc).
- ▶ Qualquer quadrado pode ser acessado individualmente através de dois índices: a linha e a coluna.

Conceituação

- ▶ A figura mostra conceitualmente o que é uma matriz.

	1	2	3	4
1	4	6	2	1
2	9	0	0	2
3	8	7	3	9
4	1	3	2	4
5	0	1	0	1

- ▶ Cada quadrado armazena exatamente uma informação de um mesmo tipo (inteiro, real, etc).
- ▶ Qualquer quadrado pode ser acessado individualmente através de dois índices: a linha e a coluna.

Conceituação

- ▶ A figura mostra conceitualmente o que é uma matriz.

	1	2	3	4
1	4	6	2	1
2	9	0	0	2
3	8	7	3	9
4	1	3	2	4
5	0	1	0	1

- ▶ Cada quadrado armazena exatamente uma informação de um mesmo tipo (inteiro, real, etc).
- ▶ Qualquer quadrado pode ser acessado individualmente através de dois índices: a linha e a coluna.

Declaração

	1	2	3	4
1	4	6	2	1
2	9	0	0	2
3	8	7	3	9
4	1	3	2	4
5	0	1	0	1

- ▶ Declaração:
`type matriz= array [1..5,1..4] of integer;`
- ▶ 1..5: índices das linhas;
- ▶ 1..4: índices das colunas;

Declaração

	1	2	3	4
1	4	6	2	1
2	9	0	0	2
3	8	7	3	9
4	1	3	2	4
5	0	1	0	1

- ▶ Outra maneira de se ver uma matriz é como sendo um vetor de vetores.

```
type vetor = array [1..4] of integer;  
    matriz = array[1..5] of vetor;  
var m: matriz;
```

Acesso

	1	2	3	4
1	4	6	2	1
2	9	0	0	2
3	8	7	3	9
4	1	3	2	4
5	0	1	0	1

► `m[linha][coluna]`

Acesso

	1	2	3	4
1	4	6	2	1
2	9	0	0	2
3	8	7	3	9
4	1	3	2	4
5	0	1	0	1

► $m[3][2]=7$

Leitura

- ▶ O laço externo “fixa” uma linha;
- ▶ Laço interno “percorre as colunas”;

[0,1], [0,2], ..., [0,10]

[1,1], [1,2], ..., [1,10]

...

[50,1], [50,2], ..., [50,10]

```
program ler_matriz;  
var w: array [0..50, 1..10] of real ;  
    i, j : integer ;  
begin  
    for i:=0 to 50 do  
        for j:=1 to 10 do  
            read (w[i,j]) ;  
        end.  
    end.
```

Impressão

```
program ler_matriz;  
var w: array [0..50, 1..10] of real ;  
    i, j : integer ;  
begin  
  for i:=0 to 50 do  
    begin  
      for j:=1 to 10 do  
        write (w[i,j], ' ');  
        writeln();  
      end;  
    end;  
end.
```

- ▶ o que deve ser alterado para imprimir somente os pares?;
- ▶ e somente aqueles com índices pares? ([2,2], [2,4], ...)?;

Impressão

```
program ler_matriz;  
var w: array [0..50, 1..10] of real ;  
    i, j : integer ;  
begin  
  for i:=0 to 50 do  
    begin  
      for j:=1 to 10 do  
        write (w[i,j], ' ');  
        writeln();  
      end;  
    end;  
end.
```

- ▶ o que deve ser alterado para imprimir somente os pares?;
- ▶ e somente aqueles com índices pares? ([2,2], [2,4], ...)?;

Impressão

```
program ler_matriz;
var w: array [0..50, 1..10] of real ;
    i, j : integer ;
begin
  for i:=0 to 50 do
    begin
      for j:=1 to 10 do
        write (w[i,j], ' ');
        writeln();
      end;
    end;
end.
```

- ▶ o que deve ser alterado para imprimir somente os pares?;
- ▶ e somente aqueles com índices pares? ([2,2], [2,4], ...)?;

Impressão da matriz transposta

▶ **Problema: Imprima a matriz por colunas (matriz transposta)**

```
program ler_matriz;  
var w: array [0..50, 1..10] of real ;  
    i, j : integer ;  
begin  
  for j:=1 to 10 do  
    begin  
      for i:=0 to 50 do  
        write (w[i,j], ' ');  
        writeln();  
      end;  
    end;  
end.
```

Impressão da matriz transposta

▶ **Problema: Imprima a matriz por colunas (matriz transposta)**

▶

```
program ler_matriz;
var w: array [0..50, 1..10] of real ;
    i, j : integer ;
begin
  for j:=1 to 10 do
  begin
    for i:=0 to 50 do
      write (w[i,j], ' ');
    writeln();
  end;
end.
```

procedimento para imprimir uma linha

▶ **Problema: Escreva um procedimento que imprime todos os elementos da linha “K” de uma matriz “w”**

▶

```
procedure imprimir_uma_linha (var w: matriz ; n,m: integer ; K: integer);  
  (* imprime a linha K da matriz *)  
  (* n eh o numero de linhas, m eh o numero de colunas *)  
  var j : integer;  
  begin  
    for j:= 1 to m do  
      write (w[K,j] ,' '); (* K fixo na primeira posicao *)  
    writeln;  
  end;
```


procedimento para imprimir uma linha

▶ **Problema: Escreva um procedimento que imprime todos os elementos da linha “K” de uma matriz “w”**

▶

```
procedure imprimir_uma_linha (var w: matriz ; n,m: integer ; K: integer);  
(* imprime a linha K da matriz *)  
(* n eh o numero de linhas, m eh o numero de colunas *)  
var j : integer;  
begin  
  for j:= 1 to m do  
    write (w[K,j] ,' '); (* K fixo na primeira posicao *)  
  writeln;  
end;
```

procedimento para imprimir uma coluna

▶ **Problema: Escreva um procedimento que imprime todos os elementos da coluna “K” de uma matriz “w”**

▶

```
procedure imprimir_uma_coluna (var w: matriz; n,m: integer; K: integer);  
  (* imprime a coluna K da matriz *)  
  (* n eh o numero de linhas, m eh o numero de colunas *)  
  var i: integer;  
  begin  
    for i:= 1 to n do  
      write (' ', w[i,K]); (* K fixo na segunda posicao *)  
    writeln;  
  end;
```

procedimento para imprimir uma coluna

▶ **Problema: Escreva um procedimento que imprime todos os elementos da coluna “K” de uma matriz “w”**

▶

```
procedure imprimir_uma_coluna (var w: matriz; n,m: integer; K: integer);  
(* imprime a coluna K da matriz *)  
(* n eh o numero de linhas, m eh o numero de colunas *)  
var i: integer;  
begin  
  for i:= 1 to n do  
    write (' ', w[i,K]); (* K fixo na segunda posicao *)  
  writeln;  
end;
```

Função para encontrar o menor elemento

▶ **Problema: Escreva uma função que retorna o menor elemento de uma matriz**

```
function acha_menor_matriz (var w: matriz ; n,m: integer) : integer;  
var i, j : integer ;  
    menor: integer ;  
begin  
    menor:= w[1,1];  
    for i:= 1 to n do  
        for j:= 1 to m do  
            if w[i,j] < menor then  
                menor:= w[i,j];  
        acha_menor_matriz:= menor;  
    end;
```

Função para encontrar o menor elemento

▶ **Problema: Escreva uma função que retorna o menor elemento de uma matriz**

```
function acha_menor_matriz (var w: matriz ; n,m: integer) : integer;  
var i, j : integer ;  
    menor: integer ;  
begin  
    menor:= w[1,1];  
    for i:= 1 to n do  
        for j:= 1 to m do  
            if w[i,j] < menor then  
                menor:= w[i,j];  
        acha_menor_matriz:= menor;  
    end;
```

Procedimento que soma duas matrizes

Problema: Escreva um procedimento que soma duas matrizes “v” e “w” em uma matriz “soma_v_w”

- ▶ “v”, “w” e “soma_v_w” devem ter o mesmo número de linhas e colunas;
- ▶ $soma_v_w := v[i,j] + w[i,j]$

```
procedure somar_matrizes (var v, w, soma_v_w: matriz ; n,m: integer);
var i, j : integer;
begin
  (* n e m sao o numero de linhas e colunas , respectivamente *)
  for i:= 1 to n do
    for j:= 1 to m do
      soma_v_w[i,j]:= v[i,j] + w[i,j];
    end;
end;
```

Procedimento que soma duas matrizes

Problema: Escreva um procedimento que soma duas matrizes “v” e “w” em uma matriz “soma_v_w”

- ▶ “v”, “w” e “soma_v_w” devem ter o mesmo número de linhas e colunas;
- ▶ $soma_v_w := v[i,j] + w[i,j]$

```
procedure somar_matrizes (var v, w, soma_v_w: matriz ; n,m: integer);
var i, j : integer;
begin
  (* n e m sao o numero de linhas e colunas , respectivamente *)
  for i:= 1 to n do
    for j:= 1 to m do
      soma_v_w[i,j]:= v[i,j] + w[i,j];
    end;
end;
```

Procedimento que soma duas matrizes

Problema: Escreva um procedimento que soma duas matrizes “v” e “w” em uma matriz “soma_v_w”

- ▶ “v”, “w” e “soma_v_w” devem ter o mesmo número de linhas e colunas;
- ▶ $soma_v_w := v[i,j] + w[i,j]$

```
procedure somar_matrizes (var v, w, soma_v_w: matriz ; n,m: integer);
var i, j : integer;
begin
  (* n e m sao o numero de linhas e colunas , respectivamente *)
  for i:= 1 to n do
    for j:= 1 to m do
      soma_v_w[i,j]:= v[i,j] + w[i,j];
    end;
end;
```


Procedimento que multiplica duas matrizes

Problema: Escreva uma procedimento que multiplica duas matrizes “A” e “B” em uma matriz “AB”

- ▶ número de colunas da matriz A tem de ser igual ao número de linhas da matriz B;
- ▶ Considere $A_{n \times m}$ e $B_{m \times p}$, então $AB_{n \times p}$

39	26	39
20	27	54
60	45	134
16	24	61
6	0	9

Matriz: AB

4	6	2	1
9	0	0	2
8	7	3	9
1	2	3	4
0	1	0	1

Matriz: A

2	3	4
5	0	0
0	7	7
1	0	9

Matriz: B

Procedimento que multiplica duas matrizes

Problema: Escreva uma procedimento que multiplica duas matrizes “A” e “B” em uma matriz “AB”

- ▶ número de colunas da matriz A tem de ser igual ao número de linhas da matriz B;
- ▶ Considere $A_{n \times m}$ e $B_{m \times p}$, então $AB_{n \times p}$

39	26	39
20	27	54
60	45	134
16	24	61
6	0	9

Matriz: AB

4	6	2	1
9	0	0	2
8	7	3	9
1	2	3	4
0	1	0	1

Matriz: A

2	3	4
5	0	0
0	7	7
1	0	9

Matriz: B

Cálculo de um elemento (B[3,2])

▶ $8 \times 3 + 7 \times 0 + 3 \times 7 + 9 \times 0 = 45$

39	26	39
20	27	54
60	45	134
16	24	61
6	0	9

▶ Matriz: AB

4	6	2	1
9	0	0	2
8	7	3	9
1	2	3	4
0	1	0	1

Matriz: A

2	3	4
5	0	0
0	7	7
1	0	9

Matriz: B

$$\begin{aligned} & A[3, 1] \times B[1, 2] + \\ & A[3, 2] \times B[2, 2] + \\ & A[3, 3] \times B[3, 2] + \\ & A[3, 4] \times B[4, 2] \end{aligned}$$

$$AB[i, j] = \sum_{k=1}^n A[i, k] \times B[k, j]$$

Cálculo de um elemento (B[3,2])

▶ $8 \times 3 + 7 \times 0 + 3 \times 7 + 9 \times 0 = 45$

39	26	39
20	27	54
60	45	134
16	24	61
6	0	9

▶ Matriz: AB

4	6	2	1
9	0	0	2
8	7	3	9
1	2	3	4
0	1	0	1

Matriz: A

2	3	4
5	0	0
0	7	7
1	0	9

Matriz: B

$A[3, 1] \times B[1, 2] +$
 $A[3, 2] \times B[2, 2] +$
 $A[3, 3] \times B[3, 2] +$
 $A[3, 4] \times B[4, 2]$

$$AB[i, j] = \sum_{k=1}^n A[i, k] \times B[k, j]$$

Cálculo de um elemento: implementação

$$AB[i,j] = \sum_{k=1}^n A[i,k] \times B[k,j]$$

- ▶ Do somatório, chega-se facilmente ao código:

```
begin (* calcula AB[i,j] *)  
  soma:= 0;  
  for k:= 1 to n do  
    soma:= soma + A[i,k] * B[k,j];  
  AB[i,j]:= soma;  
end;
```

- ▶ O último passo é calcular para todos.

Cálculo de um elemento: implementação

$$AB[i, j] = \sum_{k=1}^n A[i, k] \times B[k, j]$$

- ▶ Do somatório, chega-se facilmente ao código:

```
begin (* calcula AB[i, j] *)  
  soma:= 0;  
  for k:= 1 to n do  
    soma:= soma + A[i, k] * B[k, j];  
  AB[i, j]:= soma;  
end;
```

- ▶ O último passo é calcular para todos.

Multiplicação de Matrizes: implementação

```
procedure multiplicar_matrizes (var A: matriz;      lin_A, col_A : integer;  
                                var B: matriz;      lin_B, col_B : integer;  
                                var AB: matriz; var lin_AB, col_AB : integer);  
  
var i, j, k: integer;  
begin  
  lin_AB:= lin_A;  
  col_AB:= col_B;  
  for i:= 1 to lin_A do  
    for j:= 1 to col_B do  
      begin  
        AB[i,j]:= 0;  
        for k:= 1 to lin_B do  
          AB[i,j]:= AB[i,j] + A[i,k] * B[k,j];  
        end;  
  end;
```

- ▶ este material está no livro no capítulo 10, seções 10.1 e 10.2