

Tempo de execução de algoritmos

Multiplicação Matricial
parte 1 → algoritmo

profº Mauricio Conceição Mario

Multiplicação Matricial

Exemplo 3

Dada uma matriz A de m linhas e p colunas, e uma matriz B com p linhas e n colunas: o produto $A.B$ é a matriz de dimensão m linhas \times n colunas.

Condição: n° colunas de $A = n^\circ$ de linhas de $B \rightarrow A(p \text{ colunas}) = B(p \text{ linhas})$

$$A \cdot B = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mp} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \dots & b_{pn} \end{pmatrix} =$$

$$\begin{pmatrix} a_{11}.b_{11} + a_{12}.b_{21} + \dots + a_{1p}.b_{p1} & \dots & a_{11}.b_{1n} + a_{12}.b_{2n} + \dots + a_{1p}.b_{pn} \\ a_{21}.b_{11} + a_{22}.b_{21} + \dots + a_{2p}.b_{p1} & \dots & a_{21}.b_{1n} + a_{22}.b_{2n} + \dots + a_{2p}.b_{pn} \\ \vdots & \ddots & \vdots \\ a_{m1}.b_{11} + a_{m2}.b_{21} + \dots + a_{mp}.b_{p1} & \dots & a_{m1}.b_{1n} + a_{m2}.b_{2n} + \dots + a_{mp}.b_{pn} \end{pmatrix} = (\sum_{k=1}^p a_{ik}.b_{kj})_{m.n}$$

Exemplo 3

Multiplicação Matricial – exemplo:

Dada uma matriz A de m linhas e p colunas, e uma matriz B com p linhas e n colunas: o produto $A.B$ é a matriz de dimensão m linhas \times n colunas.

$$A = \begin{pmatrix} 5 & 8 \\ 1 & 0 \\ 2 & 7 \end{pmatrix} \quad B = \begin{pmatrix} -4 & -3 \\ 2 & 0 \end{pmatrix}$$

$$A.B = \begin{pmatrix} 5 \cdot (-4) + 8 \cdot 2 & 5 \cdot (-3) + 8 \cdot 0 \\ 1 \cdot (-4) + 0 \cdot 2 & 1 \cdot (-3) + 0 \cdot 0 \\ 2 \cdot (-4) + 7 \cdot 2 & 2 \cdot (-3) + 7 \cdot 0 \end{pmatrix} = \begin{pmatrix} -4 & -15 \\ -4 & -3 \\ 6 & -6 \end{pmatrix} 3 \times 2$$

Exemplo 3

Complexidade de Algoritmos – conceitos → **número de passos:**

→ algoritmo: **Produto de duas matrizes de dimensão $n \times n$:**

Dadas as matrizes A_{ij} e B_{ij} de dimensões $n \times n$, determinar a matriz C_{ij} igual ao produto das duas:

```
para  $i := 1, \dots, n$  fazer
  para  $j := 1, \dots, n$  fazer
    passo →  $C_{ij} := 0$ 
    para  $k := 1, \dots, n$  fazer
      passo →  $C_{ij} := C_{ij} + A_{ik} \cdot B_{kj}$ 
```

Caso de $A_{m \text{ linhas } n \text{ colunas}}$
e $B_{p \text{ linhas } q \text{ colunas}}$
→ produto = $C_{m \text{ linhas } q \text{ colunas}}$
 $k = q \text{ colunas}$

Sempre que A e B forem matrizes de mesma dimensão $n \times n$, a variável à qual a expressão matemática avaliará o tempo de execução, ou seja, a **variável independente**, é o parâmetro n . Cada passo do algoritmo corresponderá ao produto $A_{ik} \cdot B_{kj}$. O número de passos será igual ao número de produtos n . O algoritmo executa, então, n^3 passos.

(Szwarcfiter & Markenzon, 2020)

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Jul  7 10:00:53 2022
4
5  @author: cmari
6  """
7
8  import time
9  #from datetime import datetime
10
11  A = [[5, 8], [1, 0], [2, 7]]
12  B = [[-4, -3], [2, 0]]
13
14  C = [[0, 0], [0, 0], [0, 0]]
15  print("MATRIZ A:")
16  print("A[0][0] = ", A[0][0])
17  print("A[0][1] = ", A[0][1])
18  print("A[1][0] = ", A[1][0])
19  print("A[1][1] = ", A[1][1])
20  print("A[2][0] = ", A[2][0])
21  print("A[2][1] = ", A[2][1])
22  print("MATRIZ B:")
23  print("B[0][0] = ", B[0][0])
24  print("B[0][1] = ", B[0][1])
25  print("B[1][0] = ", B[1][0])
26  print("B[1][1] = ", B[1][1])
27
28  tempo_inicial = time.time()
29  for i in range(0, 3):
30      for j in range(0, 2):
31          C[i][j] = 0
32          for k in range(0, 2):
33              C[i][j] = C[i][j] + A[i][k]*B[k][j]
34  tempo_final = time.time()
35  print("PRODUTO MATRICIAL:")
36  print("MATRIZ C = ", C)
37  print("tempo de execução = ", tempo_final - tempo_inicial)
38

```

Exemplo 3

$i = 0; j = 0;$
 $C[0][0] = 0$
 $k = 0$
 $C[0][0] = A[i = 0][k = 0] * B[k=0][j=0] = 0 + 5*(-4) = -20$
 $k = 1$
 $C[0][0] = C[0][0] + A[i = 0][k = 1] * B[k=1][j=1] = -20 + 8*(2) = -4$

Console 1/A ×

```

In [6]: runfile('C:/Users/cmari/.spyder-py3/produto_matricial.py', wdir='C:/Users/
cmari/.spyder-py3')
MATRIZ A:
A[0][0] = 5
A[0][1] = 8
A[1][0] = 1
A[1][1] = 0
A[2][0] = 2
A[2][1] = 7
MATRIZ B:
B[0][0] = -4
B[0][1] = -3
B[1][0] = 2
B[1][1] = 0
PRODUTO MATRICIAL:
MATRIZ C = [[-4, -15], [-4, -3], [6, -6]]
tempo de execução = 0.0

```

$i = 0; j = 1;$
 $C[0][1] = 0$
 $k = 0$
 $C[0][1] = A[i = 0][k = 0] * B[k=0][j=1] = 0 + 5*(-3) = -15$
 $k = 1$
 $C[0][1] = C[0][1] + A[i = 0][k = 1] * B[k=1][j=1] = -15 + 8*(0) = -15$

Exemplo 3

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Jul  7 10:00:53 2022
4
5  @author: cmari
6  """
7
8  import time
9  #from datetime import datetime
10
11  A = [[5, 8], [1, 0], [2, 7]]
12  B = [[-4, -3], [2, 0]]
13
14  C = [[0, 0], [0, 0], [0, 0]]
15  print("MATRIZ A:")
16  print ("A[0][0] = ", A[0][0])
17  print ("A[0][1] = ", A[0][1])
18  print ("A[1][0] = ", A[1][0])
19  print ("A[1][1] = ", A[1][1])
20  print ("A[2][0] = ", A[2][0])
21  print ("A[2][1] = ", A[2][1])
22  print("MATRIZ B:")
23  print ("B[0][0] = ", B[0][0])
24  print ("B[0][1] = ", B[0][1])
25  print ("B[1][0] = ", B[1][0])
26  print ("B[1][1] = ", B[1][1])
27
28  tempo_inicial = time.time()
29  for p in range(1000):
30      for i in range(0, 3):
31          for j in range(0, 2):
32              C[i][j] = 0
33              for k in range(0, 2):
34                  C[i][j] = C[i][j] + A[i][k]*B[k][j]
35  tempo_final = time.time()
36  print("PRODUTO MATRICIAL:")
37  print("MATRIZ C = ", C)
38  print("tempo de execução = ", tempo_final - tempo_inicial)
    
```

Console 1/A

```

In [7]: runfile('C:/Users/cmari/.spyder-py3/produto_matricial.py', wdir='C:/Users/
cmari/.spyder-py3')
MATRIZ A:
A[0][0] = 5
A[0][1] = 8
A[1][0] = 1
A[1][1] = 0
A[2][0] = 2
A[2][1] = 7
MATRIZ B:
B[0][0] = -4
B[0][1] = -3
B[1][0] = 2
B[1][1] = 0
PRODUTO MATRICIAL:
MATRIZ C = [[-4, -15], [-4, -3], [6, -6]]
tempo de execução = 0.006857633590698242
    
```

Referências Bibliográficas

- Estruturas de Dados e Seus Algoritmos

Jayme L. Szwarcfiter & Lilian Markenzon

3ª edição – editora *gen* LTC – 2010 - 2020

- Matemática Avançada para Engenharia

Dennis G. Zill & Michael R. Cullen

Álgebra Linear e Cálculo Vetorial (2) – 3ª edição, editora Bookman - 2009