



**UNIVERSIDADE PRESBITERIANA MACKENZIE**

**FACULDADE DE COMPUTAÇÃO E INFORMÁTICA**

**ESTRUTURA DE DADOS – PROFA SOLANGE BARROS**

## FASE 1 – AULA 2

## ANÁLISE DE ALGORITMOS – MULTIPLICAÇÃO DE MATRIZES

### 1. Algoritmo padrão de multiplicação de matrizes

Além do processo tradicional de multiplicação de matrizes, existem alguns outros que foram criados visando melhorar o desempenho do algoritmo. Veja abaixo os responsáveis pela criação dos demais métodos de multiplicação de matrizes:

Inventor	Ano
Volker Strassen	1968
Victor Pan	1978
Don Coppersmith & Shmuel Winograd	1990
Andrew Stothers	2010
Virginia Williams	2011

### 2. Multiplicação de matrizes – algoritmo padrão

Dadas as matrizes  $A_{[2,2]}$  e  $B_{[2,2]}$ , calcular  $A*B$ .

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix}$$

#### ETAPA 1 - 1ª linha e 1ª coluna

$$C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 1 * (-1) + 2 * 4 & \square \\ \square & \square \end{bmatrix}$$

### ETAPA 2 - 1ª linha e 2ª coluna

$$C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 1 * 3 + 2 * 2 \\ \square & \square \end{bmatrix}$$

### ETAPA 3 - 2ª linha e 1ª coluna

$$C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 7 \\ 3 * (-1) + 4 * 4 & \square \end{bmatrix}$$

### ETAPA 4 - 2ª linha e 2ª coluna

$$C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 7 \\ 13 & 3 * 3 + 4 * 2 \end{bmatrix}$$

### RESULTADO

$$C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 7 \\ 13 & 17 \end{bmatrix}$$

3. Multiplicação de matrizes – código padrão Java

```
public class Teste1 {  
  
    public static void main (String[] args){  
        int[][] A = new int [64][64];  
        int[][] B = new int [64][64];  
        //preenche as matrizes A e B com números aleatórios  
        preencheA(A);  
        preencheB(B);  
        //cria a matriz C para receber o resultado  
        int[][] C = new int [A.length][B[0].length];  
        C = multiplica (A,B);  
        exibeResult(C);  
  
    }  
}
```

```

    public static void preencheA(int A[][]){

    }

    public static void preencheB(int B[][]){

    }

    public static int[][] multiplica (int[][]A, int[][]B) {
        int[][] C = new int [A.length][B[0].length];
        for (int i=0;i<A.length;i++)
            for (int j=0;j<B[0].length;j++){
                C[i][j]=0;
                for (int k=0;k<A[0].length;k++){
                    C[i][j]+=A[i][k] * B[k][j];
                }
            }
        return C;
    }

    public static void exhibeResult(int C[][]){

    }
}

```

#### 4. Testes

Para efeito de teste de comportamento, iremos contabilizar o custo para executar a seguinte instrução:

$$C[i][j] += A[i][k] * B[k][j];$$

Nosso objetivo é contar quantas vezes essa instrução é realizada e de que forma essa quantidade aumenta, na medida em que a matriz também aumenta de tamanho;

#### 5. Atividade para entregar

Implemente as funções que estão faltando no código acima e adicione um contador que deve ser incrementado a cada vez que a operação indicada no item 4, é executada. Sugere-se que esse contador seja uma variável global.

Para efeito de sorteio de valores no momento do preenchimento, utilize valores entre 0 e 10.

Após a exibição do resultado da multiplicação (matriz C), exiba também, a quantidade operações realizadas.

Inicialmente, faça o teste para uma matriz 2x2 e anote o resultado na tabela abaixo. Na sequência, faça outros testes, conforme indica a tabela, anotando também os resultados.

Não se preocupe com a estética para exibição da matriz resultante. Sabemos que o código está correto. O que nos interessa aqui é o processo de contagem de instruções.

<b>Tamanho da matriz</b>	<b>Número de instruções</b>
<b>2 x 2</b>	
<b>10 x 10</b>	
<b>50 x 50</b>	
<b>100 x 100</b>	
<b>200 x 200</b>	
<b>400 x 400</b>	
<b>800 x 800</b>	

#### ENTREGAR

- A tabela preenchida;
- Um gráfico de linha que mostre a curva de crescimento da execução desse algoritmo e
- Pesquise na Internet e indique a notação assintótica (BIG O) para o método tradicional de multiplicação de matrizes.

Entregue um arquivo .DOC/PDF.