

Prova de Projeto e Análise de Algoritmos

Disciplina: Projeto e Análise de Algoritmos

Instruções

Para cada uma das questões a seguir, escreva um algoritmo que resolve o problema apresentado, defina a operação básica, obtenha a expressão para o número de operações e defina a ordem de complexidade do algoritmo apresentado. Além disso, apresente exemplos de entrada e saída.

Questão 1

Problema: Escreva um algoritmo que imprima a submatriz que possui a interseção das colunas ímpares com as linhas pares de uma matriz.

Exemplo de Entrada:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Exemplo de Saída:

$$\begin{bmatrix} 2 & 4 \\ 10 & 12 \end{bmatrix}$$

Algoritmo:

Operação básica: Comparação de índices.

Complexidade: A complexidade do algoritmo é $O(m \times n)$, onde m é o número de linhas e n é o número de colunas.

Algorithm 1 Imprime colunas ímpares e linhas pares

```
1: procedure IMPRIMECOLLINPAR( $M$ )
2:   for cada linha  $i$  de  $M$  tal que  $i$  é par do
3:     for cada coluna  $j$  de  $M$  tal que  $j$  é ímpar do
4:       Imprimir  $M[i][j]$ 
5:     end for
6:   end for
7: end procedure
```

Questão 2

Problema: Escreva um algoritmo que realiza a multiplicação de duas matrizes A e B .

Multiplicação de Matrizes: Dados A de dimensão $m \times n$ e B de dimensão $n \times p$, o produto $C = A \times B$ terá dimensão $m \times p$. O elemento $C[i][j]$ é calculado como:

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

Exemplo de Entrada:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

Exemplo de Saída:

$$C = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

Algoritmo:

Operação básica: Multiplicação de elementos da matriz.

Complexidade: A complexidade do algoritmo é $O(m \times n \times p)$, onde m , n e p são as dimensões das matrizes.

Algorithm 2 Multiplicação de Matrizes

```
1: procedure MULTIPLICAMATRIZES( $A, B$ )
2:   for cada linha  $i$  de  $A$  do
3:     for cada coluna  $j$  de  $B$  do
4:        $C[i][j] \leftarrow 0$ 
5:       for cada elemento  $k$  da linha  $i$  de  $A$  do
6:          $C[i][j] \leftarrow C[i][j] + A[i][k] \times B[k][j]$ 
7:       end for
8:     end for
9:   end for
10: end procedure
```

Questão 3

Problema: Dado um array `arr` de tamanho $n - 1$ que contém inteiros distintos no intervalo de 1 a n , encontre o elemento faltante.

Exemplo de Entrada:

$$arr = [1, 2, 4, 6, 3, 7, 8]$$

Exemplo de Saída:

5

Algoritmo:

Algorithm 3 Encontra elemento faltante

```
1: procedure ENCONTRAFALTANTE( $arr, n$ )
2:    $S \leftarrow \frac{n(n+1)}{2}$  ▷ Soma dos primeiros  $n$  inteiros
3:    $S_{arr} \leftarrow \sum_{i=1}^{n-1} arr[i]$ 
4:   return  $S - S_{arr}$ 
5: end procedure
```

Operação básica: Soma de elementos.

Complexidade: A complexidade do algoritmo é $O(n)$.

Questão 4

Problema: Dado um array arr de n inteiros, encontre todos os líderes no array. Um elemento é considerado um líder se ele for maior ou igual a todos os elementos à sua direita.

Exemplo de Entrada:

$arr = [16, 17, 4, 3, 5, 2]$

Exemplo de Saída:

$[17, 5, 2]$

Algoritmo:

Algorithm 4 Encontra Líderes

```
1: procedure ENCONTRALIDERES( $arr$ )
2:    $n \leftarrow$  tamanho de  $arr$ 
3:    $max \leftarrow arr[n - 1]$ 
4:   Imprimir  $max$ 
5:   for  $i \leftarrow n - 2$  até 0 do
6:     if  $arr[i] \geq max$  then
7:       Imprimir  $arr[i]$ 
8:        $max \leftarrow arr[i]$ 
9:     end if
10:  end for
11: end procedure
```

Operação básica: Comparação entre elementos.

Complexidade: A complexidade do algoritmo é $O(n)$.