

Arrays e Matrizes

PCC104 - Projeto e Análise de Algoritmos

August 30, 2024

Instruções

1. Tente resolver cada um dos exercícios abaixo na complexidade de tempo esperada. Se não conseguir, basta apresentar uma solução qualquer.
2. Derive a complexidade de tempo e determine a classe de eficiência das soluções propostas.

1 Arrays

1. Missing in Array

Dificuldade: Fácil

Dado um array `arr` de tamanho $n-1$ que contém inteiros distintos no intervalo de 1 a n (inclusive), encontre o elemento faltante. O array é uma permutação de tamanho n com um elemento faltando. Retorne o elemento faltante.

Exemplos:

- **Entrada:** $n = 5, \text{arr} = [1, 2, 3, 5]$
Saída: 4
Explicação: Todos os números de 1 a 5 estão presentes, exceto 4.
- **Entrada:** $n = 2, \text{arr} = [1]$
Saída: 2
Explicação: Todos os números de 1 a 2 estão presentes, exceto 2.

Complexidade de Tempo Esperada: $O(n)$

Complexidade de Espaço Auxiliar Esperada: $O(1)$

2. Array Leaders

Dificuldade: Fácil

Dado um array `arr` de n inteiros positivos, sua tarefa é encontrar todos os líderes no array. Um elemento do array é considerado um líder se ele for maior que todos os elementos à sua direita ou se for igual ao elemento máximo à sua direita. O elemento mais à direita é sempre um líder.

Exemplos:

- **Entrada:** $n = 6, \text{arr} = [16, 17, 4, 3, 5, 2]$
Saída: 17 5 2
Explicação: Não há nada maior à direita de 17, 5 e 2.
- **Entrada:** $n = 5, \text{arr} = [10, 4, 2, 4, 1]$
Saída: 10 4 4 1
Explicação: Ambas as ocorrências de 4 estão na saída, pois ser igual ao maior elemento à direita também é permitido.

Complexidade de Tempo Esperada: $O(n)$

Complexidade de Espaço Auxiliar Esperada: $O(n)$

3. Second Largest

Dificuldade: Fácil

Dado um array `arr`, retorne o segundo maior elemento distinto do array. Se o segundo maior elemento não existir, retorne -1.

Exemplos:

- **Entrada:** `arr = [12, 35, 1, 10, 34, 1]`
Saída: 34
Explicação: O maior elemento do array é 35 e o segundo maior é 34.
- **Entrada:** `arr = [10, 10]`
Saída: -1
Explicação: O maior elemento do array é 10 e o segundo maior elemento não existe, então a resposta é -1.

Complexidade de Tempo Esperada: $O(n)$

Complexidade de Espaço Auxiliar Esperada: $O(1)$

4. Kadane's Algorithm

Dificuldade: Média

Dado um array de inteiros `arr[]`. Encontre o subarray contíguo (contendo pelo menos um número) que tem a soma máxima e retorne sua soma.

Exemplos:

- **Entrada:** `arr[] = [1, 2, 3, -2, 5]`
Saída: 9
Explicação: A soma máxima do subarray é 9, composta pelos elementos [1, 2, 3, -2, 5].
- **Entrada:** `arr[] = [-1, -2, -3, -4]`
Saída: -1
Explicação: A soma máxima do subarray é -1, composta pelo elemento [-1].

Complexidade de Tempo Esperada: $O(n)$

Complexidade de Espaço Auxiliar Esperada: $O(1)$

5. Indexes of Subarray Sum

Dificuldade: Média

Dado um array não ordenado `arr` de tamanho n que contém apenas inteiros não-negativos, encontre um subarray (elementos contínuos) que tenha soma igual a s . Você deve retornar os índices esquerdo e direito (indexação baseada em 1) desse subarray.

No caso de múltiplos subarrays, retorne os índices do subarray que aparece primeiro ao mover da esquerda para a direita. Se nenhum subarray existir, retorne um array contendo o elemento -1.

Exemplos:

- **Entrada:** `arr[] = [1,2,3,7,5]`, $n = 5$, $s = 12$
Saída: 2 4
Explicação: A soma dos elementos da 2ª à 4ª posição é 12.
- **Entrada:** `arr[] = [1,2,3,4,5,6,7,8,9,10]`, $n = 10$, $s = 15$ **Saída:** 1 5
Explicação: A soma dos elementos da 1ª à 5ª posição é 15.

Complexidade de Tempo Esperada: $O(n)$

Complexidade de Espaço Auxiliar Esperada: $O(1)$

2 Matrizes

1. Search in a Matrix

Dificuldade: Fácil

Dada uma matriz $\text{mat}[][]$ de tamanho $N \times M$, onde cada linha e coluna está ordenada em ordem crescente, e dado um número X . A tarefa é dizer se o elemento X está presente na matriz (1) ou não (0).

Exemplos:

- **Entrada:** $N = 3, M = 3, \text{mat}[][] = \begin{bmatrix} 3 & 30 & 38 \\ 44 & 52 & 54 \\ 57 & 60 & 69 \end{bmatrix}, X = 62$

Saída: 0

Explicação: 62 não está presente na matriz, então a saída é 0.

- **Entrada:** $N = 1, M = 6, \text{mat}[][] = [18 \ 21 \ 27 \ 38 \ 55 \ 67], X = 55$

Saída: 1

Explicação: 55 está presente na matriz na 5ª célula.

Complexidade de Tempo Esperada: $O(N + M)$

Complexidade de Espaço Auxiliar Esperada: $O(1)$

2. Transpose of Matrix

Dificuldade: Fácil

Escreva um programa para encontrar a transposta de uma matriz quadrada de tamanho $N \times N$. A transposta de uma matriz é obtida trocando-se linhas por colunas e vice-versa.

Exemplos:

- **Entrada:**

$$N = 4, \text{mat}[][] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

Saída:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

- **Entrada:** $N = 2, \text{mat}[][] = \begin{bmatrix} 1 & 2 \\ -9 & -2 \end{bmatrix}$

Saída:

$$\begin{bmatrix} 1 & -9 \\ 2 & -2 \end{bmatrix}$$

Complexidade de Tempo Esperada: $O(N \times N)$

Complexidade de Espaço Auxiliar Esperada: $O(1)$

3. Print Matrix in Snake Pattern

Dificuldade: Fácil

Dada uma matriz de tamanho $N \times N$. Imprima os elementos da matriz no padrão serpente (snake pattern) como descrito abaixo.

1	2	3	4
8	7	6	5
9	10	11	12
16	15	14	13

Exemplos:

- **Entrada:** $N = 3, \text{mat}[][] = \begin{bmatrix} 45 & 48 & 54 \\ 21 & 89 & 87 \\ 70 & 78 & 15 \end{bmatrix}$

Saída: 45 48 54 87 89 21 70 78 15

Explicação: A matriz é como abaixo:

$$\begin{bmatrix} 45 & 48 & 54 \\ 21 & 89 & 87 \\ 70 & 78 & 15 \end{bmatrix}$$

Imprimindo no padrão serpente, temos a saída como 45 48 54 87 89 21 70 78 15.

- **Entrada:** $N = 2, \text{mat}[][] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

Saída: 1 2 4 3

Explicação: A matriz é como abaixo:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Imprimindo no padrão serpente, temos a saída como 1 2 4 3.

Complexidade de Tempo Esperada: $O(N \times N)$

Complexidade de Espaço Auxiliar Esperada: $O(N \times N)$ apenas para a lista resultante.

4. Row with Minimum Number of 1's

Dificuldade: Fácil

Dada uma matriz binária 2D (indexada a partir de 1) de dimensões $n \times m$, determine a linha que contém o menor número de 1's.

Nota: A matriz contém apenas 1's e 0's. Se duas ou mais linhas contiverem o menor número de 1's, a resposta é o menor desses índices.

Exemplos:

- **Entrada:** $n = 4, m = 4, \text{mat}[][] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

Saída: 2

Explicação: As linhas 2 e 3 contêm o menor número de 1's (2 cada). Como a linha 2 é menor que a linha 3, a resposta é 2.

- **Entrada:** $n = 3, m = 3, \text{mat}[][] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Saída: 1

Explicação: Todas as linhas contêm o mesmo número de 1's (0 cada). Entre elas, o índice 1 é o menor, então a resposta é 1.

Complexidade de Tempo Esperada: $O(n \times m)$

Complexidade de Espaço Auxiliar Esperada: $O(1)$

5. Rotate by 90 Degree

Dificuldade: Média

Dada uma matriz quadrada $\text{mat}[][]$ de tamanho $N \times N$. A tarefa é rotacioná-la em 90 graus no sentido anti-horário sem usar nenhum espaço extra.

Exemplos:

- **Entrada:** $N = 3, \text{mat}[][] = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

Saída:

$$\begin{bmatrix} 3 & 6 & 9 \\ 2 & 5 & 8 \\ 1 & 4 & 7 \end{bmatrix}$$

Complexidade de Tempo Esperada: $O(N \times N)$

Complexidade de Espaço Auxiliar Esperada: $O(1)$

6. Find Whether Path Exists

Dificuldade: Média

Dada uma grade de tamanho $n \times n$ preenchida com 0, 1, 2, 3. Verifique se há um caminho possível do ponto de origem ao destino. Você pode percorrer para cima, para baixo, para a direita e para a esquerda.

Descrição das Células:

- Um valor de célula 1 significa Origem.
- Um valor de célula 2 significa Destino.
- Um valor de célula 3 significa célula vazia.
- Um valor de célula 0 significa Parede (célula bloqueada que não pode ser atravessada).

Nota: Há apenas uma única origem e um único destino.

Exemplos:

- **Entrada:** $\text{grid} = \begin{bmatrix} 3 & 0 & 3 & 0 & 0 \\ 3 & 0 & 0 & 0 & 3 \\ 3 & 3 & 3 & 3 & 3 \\ 0 & 2 & 3 & 0 & 0 \\ 3 & 0 & 0 & 1 & 3 \end{bmatrix}$

Saída: 0

Explicação: A grade é como abaixo:

$$\begin{bmatrix} 3 & 0 & 3 & 0 & 0 \\ 3 & 0 & 0 & 0 & 3 \\ 3 & 3 & 3 & 3 & 3 \\ 0 & 2 & 3 & 0 & 0 \\ 3 & 0 & 0 & 1 & 3 \end{bmatrix}$$

Não há caminho para chegar ao destino (3,1) a partir da origem (4,3).

- **Entrada:** $\text{grid} = \begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix}$

Saída: 1

Explicação: A grade é como abaixo:

$$\begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix}$$

Há um caminho da origem (0,0) ao destino (1,1).

Complexidade de Tempo Esperada: $O(n^2)$

Complexidade de Espaço Auxiliar Esperada: $O(n^2)$