

Arrays, Matrizes, Strings e Listas Encadeadas

PCC104 - Projeto e Análise de Algoritmos

August 14, 2024

Exercícios

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:** $\text{arr} = [12, 35, 1, 10, 34, 1]$
Saída: 34
Explicação: O maior elemento do array é 35 e o segundo maior é 34.
- **Entrada:** $\text{arr} = [10, 10]$
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 $\text{arr}[]$. Encontre o subarray contíguo (contendo pelo menos um número) que tem a soma máxima e retorne sua soma.

Exemplos:

- **Entrada:** $\text{arr}[] = [1, 2, 3, -2, 5]$
Saída: 9
Explicação: A soma máxima do subarray é 9, composta pelos elementos $[1, 2, 3, -2, 5]$.
- **Entrada:** $\text{arr}[] = [-1, -2, -3, -4]$
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:** $\text{arr}[] = [1, 2, 3, 7, 5]$, $n = 5$, $s = 12$
Saída: 2 4
Explicação: A soma dos elementos da 2ª à 4ª posição é 12.
- **Entrada:** $\text{arr}[] = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, $n = 10$, $s = 15$ **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: $O(1)$