

Etapa 2: Desafio Técnico

Instruções:

- O uso de inteligências artificiais para resolução do teste não é autorizado.
- A linguagem de programação é de livre escolha do candidato.
- Não conseguir responder todas as questões não significa reprovação imediata. No entanto, os resultados serão ranqueados, pois temos um número limitado de vagas.
- Demonstrar criatividade na resolução dos problemas, como propor abordagens diferentes para estruturas de dados, algoritmos otimizados ou soluções bem organizadas e documentadas, poderá contribuir positivamente para sua avaliação.
- A entrega deverá ser feita em um repositório Git, cujo link deverá ser enviado por e-mail (as instruções detalhadas sobre envio serão encaminhadas em breve).

1.Soma de Dois Números

Descrição:

Dado um array de inteiros **nums** e um inteiro **target**, retorne os índices dos dois números que somam o valor **target**.

Você pode assumir que cada entrada terá **exatamente uma** solução, e você **não pode** usar o mesmo elemento duas vezes.

Você pode retornar a resposta em qualquer ordem.

Exemplos:

Entrada: nums = [2,7,11,15], target = 9

Saída: [0,1]

Explicação: Porque $\text{nums}[0] + \text{nums}[1] == 9$, retornamos [0, 1].

Entrada: nums = [3,2,4], target = 6

Saída: [1,2]

Entrada: nums = [3,3], target = 6

Saída: [0,1]

Restrições:

- $2 \leq \text{nums.length} \leq 10^4$
- $-10^9 \leq \text{nums}[i] \leq 10^9$
- $-10^9 \leq \text{target} \leq 10^9$
- Existe apenas uma resposta válida.

2.Verificador de Palíndromo

Descrição:

Uma frase é considerada um palíndromo se, após converter todas as letras maiúsculas em minúsculas e remover todos os caracteres não alfanuméricos, ela for lida da mesma forma da esquerda para a direita e vice-versa. Caracteres alfanuméricos incluem letras e números.

Dada uma string **s**, retorne **true** se ela for um palíndromo, ou **false** caso contrário.

Exemplos:

Entrada: s = "A man, a plan, a canal: Panama"

Saída: true

Explicação: "amanaplanacanalpanama" é um palíndromo.

Entrada: `s = "race a car"`

Saída: `false`

Explicação: "raceacar" não é um palíndromo.

Entrada: `s = " "`

Saída: `true`

Explicação: Após remover os caracteres não alfanuméricos, **s** torna-se uma string vazia `""`. Como uma string vazia lida igual nos dois sentidos, é um palíndromo.

Restrições:

- $1 \leq s.length \leq 2 * 10^5$
 - **s** consiste apenas de caracteres ASCII imprimíveis.
-

3. FizzBuzz

Descrição:

Dado um número inteiro **n**, **retorne** um array de strings resposta (indexado a partir de 1), onde:

- `resposta[i] == "FizzBuzz"` se **i** for divisível por 3 e 5.
- `resposta[i] == "Fizz"` se **i** for divisível apenas por 3.
- `resposta[i] == "Buzz"` se **i** for divisível apenas por 5.
- `resposta[i] == i` (como string) se nenhuma das condições acima for satisfeita.

Exemplos:

Entrada: `n = 3`

Saída: `["1","2","Fizz"]`

Entrada: `n = 5`

Saída: `["1","2","Fizz","4","Buzz"]`

Entrada: `n = 15`

Saída: `["1","2","Fizz","4","Buzz","Fizz","7","8","Fizz","Buzz","11","Fizz","13","14","FizzBuzz"]`

4. Maximum Subarray (Subarray com Maior Soma)

Descrição:

Dado um array de inteiros nums, encontre o subarray contíguo com a maior soma e retorne essa soma.

Exemplos:

Entrada: `[-2, 1, -3, 4, -1, 2, 1, -5, 4]` → Saída: 6

Entrada: `[1]` → Saída: 1

Entrada: `[5, 4, -1, 7, 8]` → Saída: 23

Restrições:

- $1 \leq \text{nums.length} \leq 10^5$
- $-10^4 \leq \text{nums}[i] \leq 10^4$

5. Best Time to Buy and Sell Stock (Melhor Momento para Comprar e Vender Ações)

Descrição:

Dado um array prices onde `prices[i]` é o preço de uma ação no i-ésimo dia, encontre o lucro máximo possível comprando e vendendo uma vez.

Exemplos:

Entrada: `[7, 1, 5, 3, 6, 4]` → Saída: 5

Entrada: `[7, 6, 4, 3, 1]` → Saída: 0

Restrições:

- $1 \leq \text{prices.length} \leq 10^5$
- $0 \leq \text{prices}[i] \leq 10^4$

Conceitos: Rastreamento de mínimos, cálculo de máximos

6. Climbing Stairs (Subindo Escadas)

Descrição:

Você está subindo uma escada com n degraus, podendo subir 1 ou 2 de cada vez.
De quantas formas diferentes é possível chegar ao topo?

Exemplos:

Entrada: $n = 2 \rightarrow$ Saída: 2

Entrada: $n = 3 \rightarrow$ Saída: 3

Restrições:

- $1 \leq n \leq 45$

Conceitos: Programação dinâmica, sequência de Fibonacci

7. Valid Parentheses

Descrição:

Dada uma string s contendo apenas os caracteres '(', ')', '{', '}', '[' e ']', determine se a string é válida.

Uma string de entrada é considerada **válida** se:

- Parênteses abertos forem fechados pelo mesmo tipo de parêntese;
- Parênteses abertos forem fechados na ordem correta;
- Todo parêntese de fechamento tiver um correspondente de abertura do mesmo tipo.

Exemplos:

Entrada: $s = "()"$

Saída: true

Entrada: $s = "()[]\{\}"$

Saída: true

Entrada: $s = "("$

Saída: false

Entrada: $s = "([])"$

Saída: true

Entrada: $s = "([)]"$

Saída: false

Restrições:

- $1 \leq s.length \leq 10^4$
 - A string **s** contém apenas os caracteres '(', ')', '{', '}', '[', '']'
-

8. Merge Intervals**Descrição:**

Dado um array de intervalos onde **intervals[i] = [starti, endi]**, una todos os intervalos que se sobrepõem e retorne um novo array apenas com os intervalos não sobrepostos que cobrem todos os intervalos da entrada.

Exemplos:

Entrada: intervals = [[1,3],[2,6],[8,10],[15,18]]

Saída: [[1,6],[8,10],[15,18]]

Explicação: Os intervalos [1,3] e [2,6] se sobrepõem e são unidos em [1,6].

Entrada: intervals = [[1,4],[4,5]]

Saída: [[1,5]]

Explicação: Os intervalos [1,4] e [4,5] são considerados sobrepostos.

Restrições:

- $1 \leq intervals.length \leq 10^4$
 - $intervals[i].length == 2$
 - $0 \leq starti \leq endi \leq 10^4$
-

9. Missing Number**Descrição:**

Dado um array **nums** contendo **n** números distintos dentro do intervalo **[0, n]**, retorne o **único número** que está faltando no intervalo.

Exemplos:

Entrada: nums = [3,0,1]

Saída: 2

Explicação: $n = 3$, então os números esperados são [0,1,2,3]. O número 2 está faltando.

Entrada: nums = [0,1]

Saída: 2

Explicação: $n = 2$, então os números esperados são $[0, 1, 2]$. O número 2 está faltando.

Entrada: `nums = [9, 6, 4, 2, 3, 5, 7, 0, 1]`

Saída: 8

Explicação: $n = 9$, então os números esperados são $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$. O número 8 está faltando.

Restrições:

- $n == \text{nums.length}$
- $1 \leq n \leq 10^4$
- $0 \leq \text{nums}[i] \leq n$
- Todos os números de **nums** são únicos

Follow-up: Consegue implementar uma solução com complexidade de tempo $O(n)$ e usando apenas $O(1)$ de espaço extra?

10. Median of Two Sorted Arrays

Descrição:

Dado dois arrays ordenados **nums1** e **nums2**, de tamanhos **m** e **n** respectivamente, retorne a **mediana** dos dois arrays ordenados combinados.

A complexidade geral da solução **deve ser** $O(\log(m+n))$.

Exemplos:

Entrada: `nums1 = [1, 3]`, `nums2 = [2]`

Saída: 2.00000

Explicação: array mesclado = `[1, 2, 3]`, e a mediana é 2.

Entrada: `nums1 = [1, 2]`, `nums2 = [3, 4]`

Saída: 2.50000

Explicação: array mesclado = `[1, 2, 3, 4]`, e a mediana é $(2 + 3) / 2 = 2.5$.

Restrições:

- `nums1.length == m`

- `nums2.length == n`
 - `0 <= m <= 1000`
 - `0 <= n <= 1000`
 - `1 <= m + n <= 2000`
 - `-106 <= nums1[i], nums2[i] <= 106`
-