Preliminares - Algoritmos

Estrutura de Dados

Professor: Henrique Viana Oliveira, henriq.viana@uece.br

Construindo Algoritmos

Exercício: 1.

(Número Faltante) Dado um vetor *nums* contendo n números distintos no intervalo $[0, n]$, retorna o único número no intervalo que está faltando no array.

Exemplo 1. Entrada: $nums = [3, 0, 1]$. **Saída:** 2. **Explicação:** $n = 3$, pois há 3 números, então todos os números estão no intervalo [0,3]. 2 é o número que falta no intervalo, pois não aparece em *nums*.

Exemplo 2. Entrada: $nums = [0, 1]$. **Saída:** 2. **Explicação:** $n = 2$, pois há 2 números, então todos os números estão no intervalo [0,2]. 2 é o número que falta no intervalo, pois não aparece em *nums*.

Exemplo 3. Entrada: $nums = [9, 6, 4, 2, 3, 5, 7, 0, 1]$. **Saída:** 8. **Explicação:** $n = 9$, pois há 9 números, então todos os números estão no intervalo [0,9]. 8 é o número que falta no intervalo, pois não aparece em *nums*.

Exercício: 2.

(Duas Somas) Dado um vetor de números inteiros *nums* e um *alvo* inteiro, retorne os índices dos dois números de forma que a soma deles resulte no *alvo*. Você pode assumir que cada entrada teria exatamente uma solução e não pode usar o mesmo elemento duas vezes. Você pode retornar a resposta em qualquer ordem.

Exemplo 1. Entrada: $nums = [2, 7, 11, 15]$, $alvo = 9$. **Saída:** [0,1]. **Explicação:** Como $nums[0] + nums[1] == 9$, retornamos [0, 1].

Exemplo 2. Entrada: $nums = [3, 2, 4]$, $alvo = 6$. **Saída:** [1,2].

Exemplo 3. Entrada: $nums = [3, 3]$, $alvo = 6$. **Saída:** [0,1].

Exercício: 3.

(Contém Duplicado) Dado um vetor de inteiros *nums*, retorna *true* se qualquer valor aparecer pelo menos duas vezes no vetor, e retorna *false* se cada elemento for distinto.

Exemplo 1. Entrada: $nums = [1, 2, 3, 1]$. **Saída:** *true*. **Explicação:** O elemento 1 ocorre nos índices 0 e 3.

Exemplo 2. Entrada: $nums = [1, 2, 3, 4]$. **Saída:** *false*. **Explicação:** Todos os elementos são distintos.

Exemplo 3. Entrada: $nums = [1, 1, 1, 3, 3, 4, 3, 2, 4, 2]$. **Saída:** *true*.

Exercício: 4.

(Número Único) Dado um vetor não vazio de números inteiros, cada elemento aparece duas vezes, exceto um. Encontre esse único elemento.

Exemplo 1. Entrada: $nums = [2, 2, 1]$. **Saída:** 1.

Exemplo 2. Entrada: $nums = [4, 1, 2, 1, 2]$. **Saída:** 4.

Exemplo 3. Entrada: $nums = [1]$. **Saída:** 1.

Exercício: 5.

(Mais um) Você recebe um inteiro grande representado como um vetor de dígitos inteiros, onde cada $digitos[i]$ é o i-ésimo dígito do inteiro. Os dígitos são ordenados do mais significativo para o menos significativo, da esquerda para a direita. O inteiro grande não contém nenhum 0 à esquerda. Incremente o inteiro grande em um e retorne o vetor de dígitos resultante.

Exemplo 1. Entrada: $digitos = [1, 2, 3]$. **Saída:** [1,2,4]. **Explicação:** O vetor representa o inteiro 123. Incrementando em uma unidade, obtém-se 123 + 1 = 124. Portanto, o resultado deve ser [1,2,4].

Exemplo 2. Entrada: $digitos = [4, 3, 2, 1]$. **Saída:** [4,3,2,2]. **Explicação:** O vetor representa o inteiro 4321. Incrementando em uma unidade, obtém-se 4321 + 1 = 4322. Portanto, o resultado deve ser [4,3,2,2].

Exemplo 3. Entrada: $digitos = [9]$. **Saída:** [1,0]. **Explicação:** O vetor representa o inteiro 9. Incrementando em um, obtém-se 9 + 1 = 10. Portanto, o resultado deve ser [1,0].

Exercício: 6.

(Mover Zeros) Dado um vetor de inteiros $nums$, mova todos os 0s para o final dele, mantendo a ordem relativa dos elementos diferentes de zero. Observe que você deve fazer isso no local, sem fazer uma cópia do vetor.

Exemplo 1. Entrada: $nums = [0, 1, 0, 3, 12]$. **Saída:** [1,3,12,0,0].

Exemplo 2. Entrada: $nums = [0]$. **Saída:** [0].

Exercício: 7.

(Interseção de Dois Vetores) Dados dois vetores de inteiros, $nums1$ e $nums2$, retorne um vetor de suas interseções. Cada elemento no resultado deve ser único e você pode retornar o resultado em qualquer ordem.

Exemplo 1. Entrada: $nums1 = [1, 2, 2, 1]$, $nums2 = [2, 2]$. **Saída:** [2].

Exemplo 2. Entrada: $nums1 = [4, 9, 5]$, $nums2 = [9, 4, 9, 8, 4]$. **Saída:** [9,4]. **Explicação:** [4,9] também é aceito.

Exercício: 8.

(Troco da Limonada) Em uma barraca de limonada, cada limonada custa R$ 5. Os clientes estão em uma fila para comprar de você e pedir uma limonada de cada vez (na ordem especificada nas notas). Cada cliente comprará apenas uma limonada e pagará com uma nota de R$ 5, R$ 10 ou R$ 20. Você deve fornecer o troco correto a cada cliente para que a transação líquida seja de R$ 5. Observe que você não tem troco em mãos inicialmente. Dado um vetor de inteiros *contas*, onde $contas[i]$ é a conta que o i-ésimo cliente paga, retorne *true* se você puder fornecer a cada cliente o troco correto, ou *false* caso contrário.

Exemplo 1. Entrada: $contas = [5, 5, 5, 10, 20]$. **Saída:** *true*. **Explicação:** Dos 3 primeiros clientes, coletamos três notas de R$ 5 em ordem. Do quarto cliente, coletamos uma nota de R$ 10 e devolvemos uma nota de R$ 5. Do quinto cliente, coletamos uma nota de R$ 10 e uma nota de R$ 5. Como todos os clientes receberam o troco correto, a saída é *true*.

Exemplo 2. Entrada: $contas = [5, 5, 10, 10, 20]$. **Saída:** *false*. **Explicação:** Dos dois primeiros clientes em ordem, coletamos duas notas de R$ 5. Para os dois clientes seguintes em ordem, coletamos uma nota de R$ 10 e devolvemos uma nota de R$ 5. Para o último cliente, não podemos devolver o troco de R$ 15 porque temos apenas duas notas de R$ 10. Como nem todos os clientes receberam o troco correto, a resposta é *false*.

Exercício: 9.

(Último Peso de Pedra) Você recebe um vetor de pedras inteiras, onde $pedras[i]$ é o peso da i-ésima pedra. Estamos jogando um jogo com as pedras. Em cada jogada, escolhemos as duas pedras mais pesadas e as quebramos. Suponha que as duas pedras mais pesadas tenham pesos x e y, com $x \leq y$. O resultado dessa quebra é:

- Se $x = y$, ambas as pedras são destruídas, e
- Se $x \neq y$, a pedra de peso x é destruída, e a pedra de peso y tem um novo peso $y - x$.

No final do jogo, resta no máximo uma pedra. Retorna o peso da última pedra restante. Se não houver mais pedras, retorna 0.

Exemplo 1. Entrada: $pedras = [2, 7, 4, 1, 8, 1]$. **Saída:** 1. **Explicação:**

Combinamos 7 e 8 para obter 1, então o vetor se converte em [2,4,1,1,1] e,

combinamos 2 e 4 para obter 2, então o vetor se converte em [2,1,1,1] e,

combinamos 2 e 1 para obter 1, então o vetor se converte em [1,1,1] e,

combinamos 1 e 1 para obter 0, então o vetor se converte em [1], então esse é o valor da última pedra.

Exemplo 2. Entrada: $pedras = [1]$. **Saída:** 1.

Exercício: 10.

(Ordenar Vetor por Paridade) Dado um vetor de inteiros $nums$, mova todos os inteiros pares no início do vetor, seguidos por todos os inteiros ímpares. Retorne qualquer array que satisfaça esta condição.

Exemplo 1. Entrada: $nums = [3, 1, 2, 4]$. **Saída:** [2,4,3,1]. **Explicação:** As saídas [4,2,3,1], [2,4,1,3] e [4,2,1,3] também seriam aceitas.

Exemplo 2. Entrada: $nums = [0]$. **Saída:** [0].