

Fehlende Zahl

Aufgabenstellung

Gegeben ist ein Array `nums`, das `n` unterschiedliche Zahlen im Bereich `[0, n]` enthält. Es fehlt genau eine Zahl aus diesem Bereich im Array. Schreiben Sie eine Funktion, die die fehlende Zahl findet und zurückgibt.

Beispiele

1. Eingabe: `nums = [3,0,1]` Ausgabe: 2 Erklärung: `n = 3`, da es 3 Zahlen gibt. 2 fehlt im Array.
2. Eingabe: `nums = [0,1]` Ausgabe: 2 Erklärung: `n = 2`, da es 2 Zahlen gibt. 2 fehlt im Array.
3. Eingabe: `nums = [9,6,4,2,3,5,7,0,1]` Ausgabe: 8 Erklärung: `n = 9`, da es 9 Zahlen gibt. 8 fehlt im Array.
4. Eingabe: `nums = [1]` Ausgabe: 0 Erklärung: `n = 1`, da es 1 Zahl gibt. 0 fehlt im Array.
5. Eingabe: `nums = [0]` Ausgabe: 1 Erklärung: `n = 1`, da es 1 Zahl gibt. 1 fehlt im Array.
6. Eingabe: `nums = [4,3,2,1,0,6]` Ausgabe: 5 Erklärung: `n = 6`, da es 6 Zahlen gibt. 5 fehlt im Array.
7. Eingabe: `nums = [8,6,4,2,3,5,7,0,1]` Ausgabe: 9 Erklärung: `n = 9`, da es 9 Zahlen gibt. 9 fehlt im Array.
8. Eingabe: `nums = [5,3,2,1,0,4,7]` Ausgabe: 6 Erklärung: `n = 7`, da es 7 Zahlen gibt. 6 fehlt im Array.

Randbedingungen

- `n == nums.length`
- `1 <= n <= 10^4`
- `0 <= nums[i] <= n`
- Alle Zahlen in `nums` sind eindeutig.

Lösungsvorschlag (Pseudocode)

```
FUNCTION findMissingNumber(nums):  
    n = LENGTH(nums)  
    expectedSum = n * (n + 1) / 2  
    actualSum = 0  
  
    FOR i FROM 0 TO n - 1:  
        actualSum = actualSum + nums[i]  
  
    missingNumber = expectedSum - actualSum  
  
    RETURN missingNumber  
  
END FUNCTION
```

Dieser Pseudocode verwendet die Gauß'sche Summenformel, um die erwartete Summe aller Zahlen von 0 bis n zu berechnen. Dann wird die tatsächliche Summe der Zahlen im Array berechnet. Die Differenz zwischen der erwarteten und der tatsächlichen Summe ergibt die fehlende Zahl.

Vorgehen

- Projekt auf GitHub anlegen
- dieses Dokument im Projekt aufnehmen
- Lösung in JavaScript erarbeiten
- alle Test aus dem Dokument durchführen und Ergebnisse vergleichen
- Zeitaufwand 90 Minuten
-