3432. Count Partitions with Even Sum Difference

Solved

Easy | Topics | Companies | Hint

You are given an integer array `nums` of length `n`.

A **partition** is defined as an index `i` where `0 <= i < n - 1`, splitting the array into two **non-empty** subarrays such that:

- Left subarray contains indices `[0, i]`.
- Right subarray contains indices `[i + 1, n - 1]`.

Return the number of **partitions** where the **difference** between the **sum** of the left and right subarrays is **even**.

Example 1:

> **Input:** `nums = [10,10,3,7,6]`
>
> **Output:** 4
>
> **Explanation:**
>
> The 4 partitions are:
>
> - `[10]`, `[10, 3, 7, 6]` with a sum difference of `10 - 26 = -16`, which is even.
> - `[10, 10]`, `[3, 7, 6]` with a sum difference of `20 - 16 = 4`, which is even.
> - `[10, 10, 3]`, `[7, 6]` with a sum difference of `23 - 13 = 10`, which is even.
> - `[10, 10, 3, 7]`, `[6]` with a sum difference of `30 - 6 = 24`, which is even.

Example 2:

> **Input:** `nums = [1,2,2]`
>
> **Output:** `0`
>
> **Explanation:**
>
> No partition results in an even sum difference.

Example 3:

> **Input:** nums = [2,4,6,8]
>
> **Output:** 3
>
> **Explanation:**
>
> All partitions result in an even sum difference.

Constraints:

- `2 <= n == nums.length <= 100`
- `1 <= nums[i] <= 100`

Python:

```
class Solution:
    def countPartitions(self, nums):
        total_sum = sum(nums)
        left_sum = 0
        count = 0

        for i in range(len(nums) - 1):
            left_sum += nums[i]
            right_sum = total_sum - left_sum


            if (left_sum % 2) == (right_sum % 2):
                count += 1

        return count
```

JavaScript:

```
const countPartitions = A =>
    A.reduce((a, c) => a + c, 0) & 1 ? 0 : A.length - 1;
```

Java:

```
class Solution {
```

```
public int countPartitions(int[] nums) {
    int totalSum = 0;
    for (int i=0;i<nums.length;i++) {
        totalSum += nums[i];
    }

    int leftSum = 0;
    int count = 0;

    for (int i = 0; i < nums.length - 1; i++) {
        leftSum += nums[i];
        int rightSum = totalSum - leftSum;

        if ((leftSum % 2) == (rightSum % 2)) {
            count++;
        }
    }
    return count;
}
}
```