LeetCode 2574 - Left and Right Sum Differences

Problem Understanding

We are given an array nums of size n.

We must find:

- leftSum[i] = sum of all elements before index i
- rightSum[i] = sum of all elements after index i
- answer[i] = |leftSum[i] - rightSum[i]|

Brute Force Approach

Idea:

For each index i:

- Loop left from 0 to i-1 to find leftSum[i]
- Loop right from i+1 to n-1 to find rightSum[i]
- Compute |leftSum[i] - rightSum[i]|

Time Complexity: O(n^2)

Dry Run (nums = [10, 4, 8, 3])

i | leftSum | rightSum | |leftSum - rightSum|

---

0 | 0 | 15 | 15

1 | 10 | 11 | 1

2 | 14 | 3 | 11

3 | 22 | 0 | 22

Output: [15, 1, 11, 22]

Brute Force Code (Python, Java, C++) provided above.

Optimized Approach (Prefix Sum)

Idea:

Precompute:

- leftSum[i] = leftSum[i-1] + nums[i-1]
- rightSum[i] = rightSum[i+1] + nums[i+1]

Then answer[i] = |leftSum[i] - rightSum[i]|

Time: O(n), Space: O(n)

Dry Run (nums = [10, 4, 8, 3])

leftSum = [0, 10, 14, 22]

rightSum = [15, 11, 3, 0]

answer = [15, 1, 11, 22]

Summary

Approach	Time	Space	Explanation
Brute Force	O(n^2)	O(1)	Compute left & right sum each time
Prefix Sum	O(n)	O(n)	Precompute prefix & suffix arrays