

Blind 75: Product of Array Except Self

Category: Array Manipulation

DIFFICULTY: MEDIUM

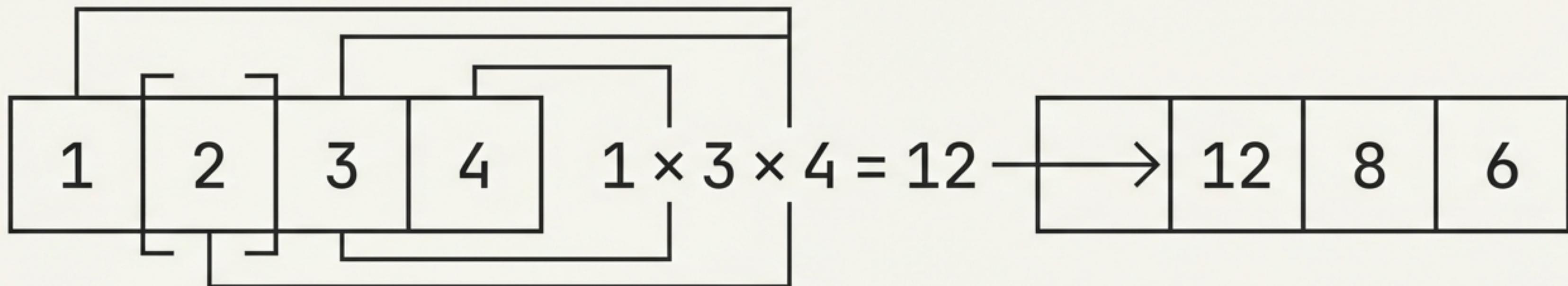
The Objective

Input

1	2	3	4
---	---	---	---

Output

24	12	8	6
----	----	---	---



Target index 1: Ignore "2". Multiply the rest.

The Constraints Force the Pattern

Time Complexity: $O(n)$



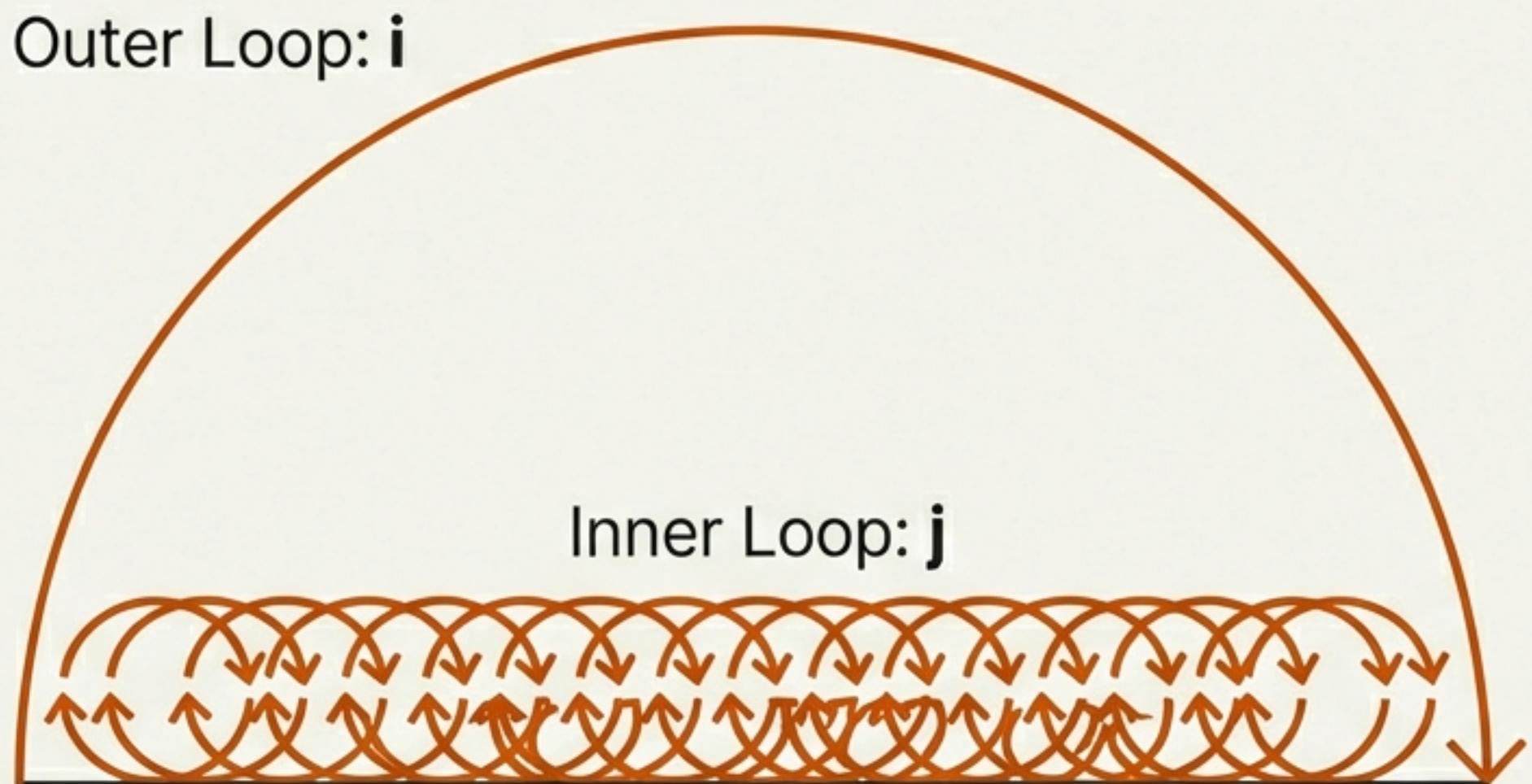
Linear time required. No nested loops.

No Division

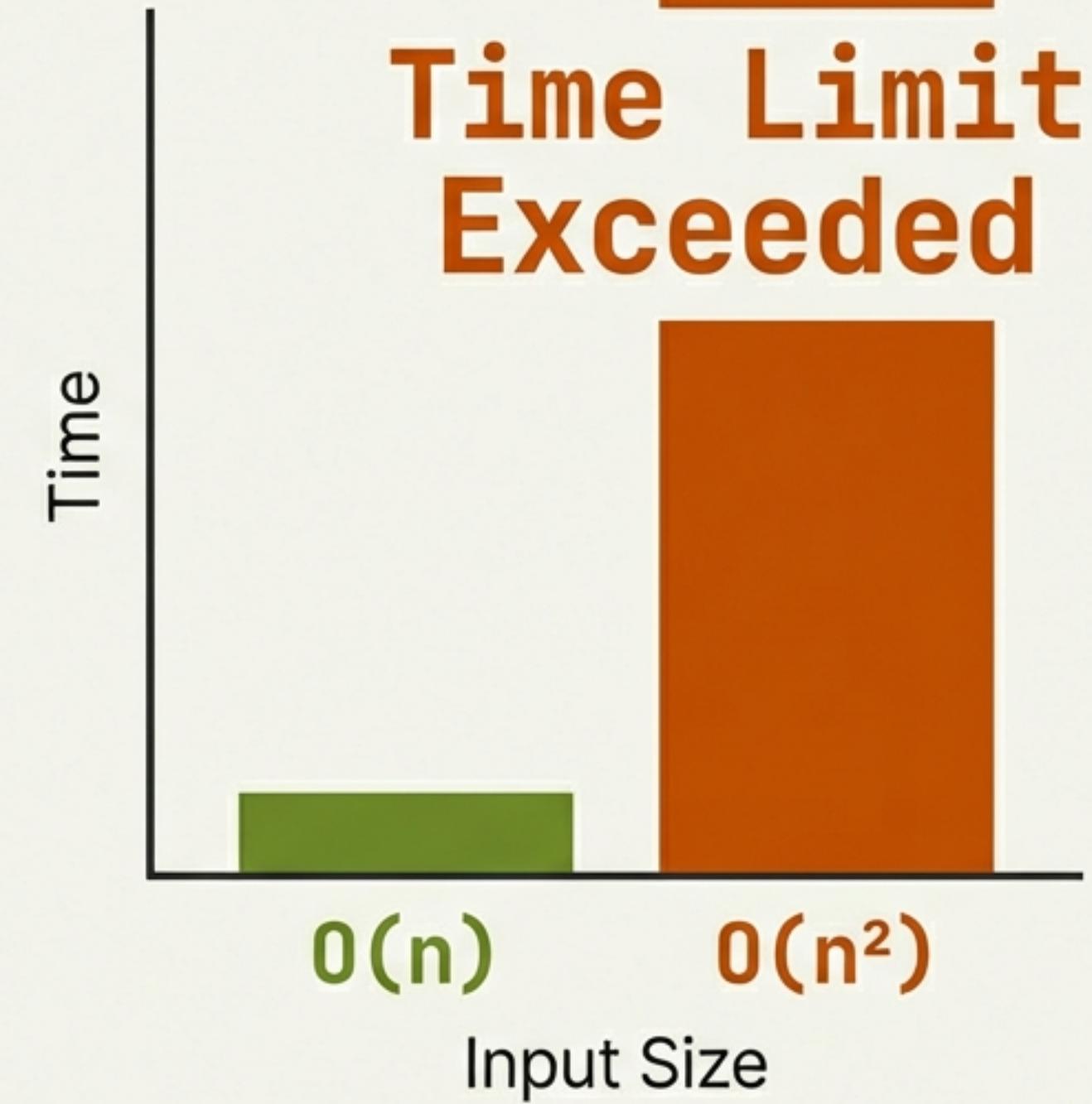


Cannot divide total product by current element.

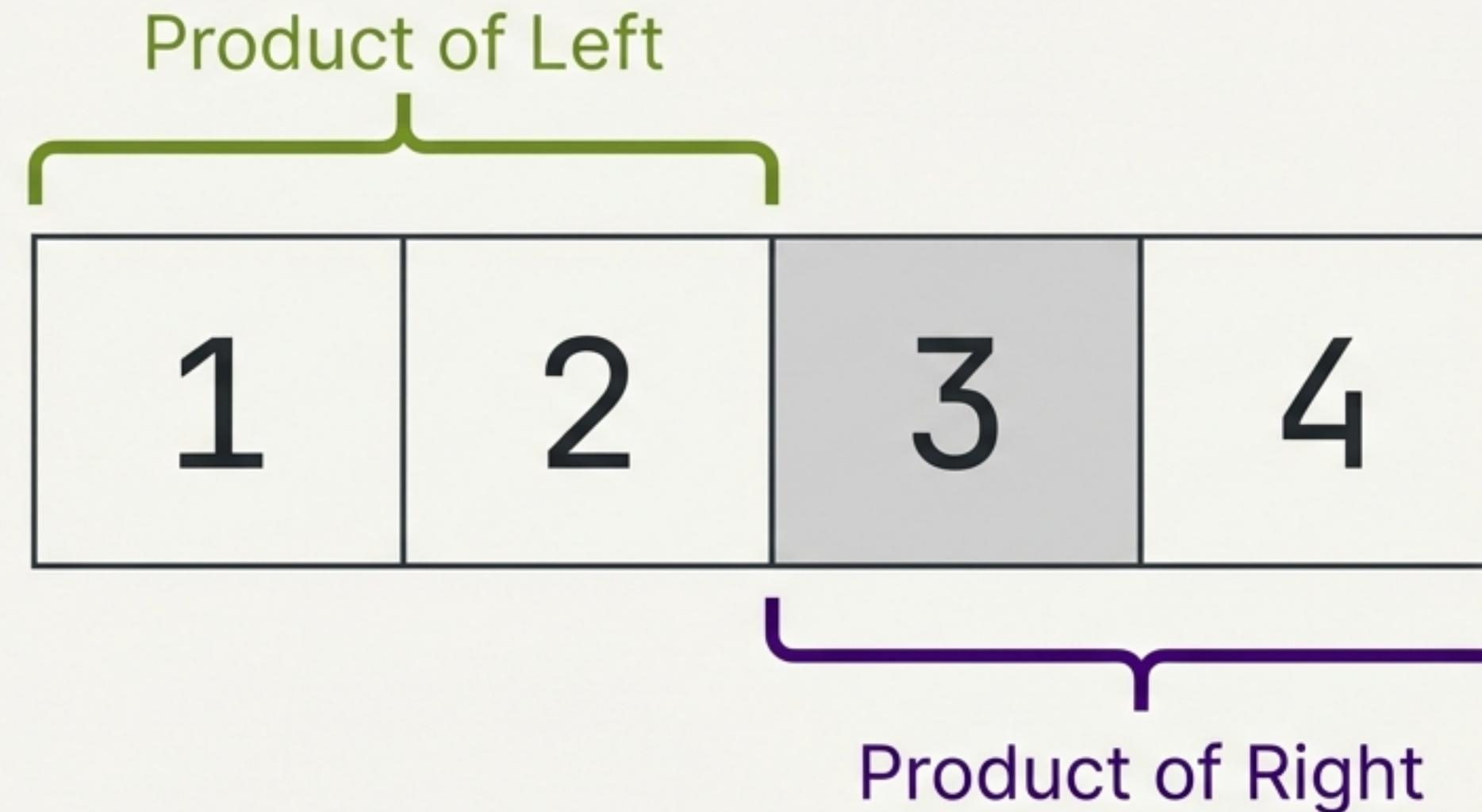
The Brute Force Trap



$O(n^2)$

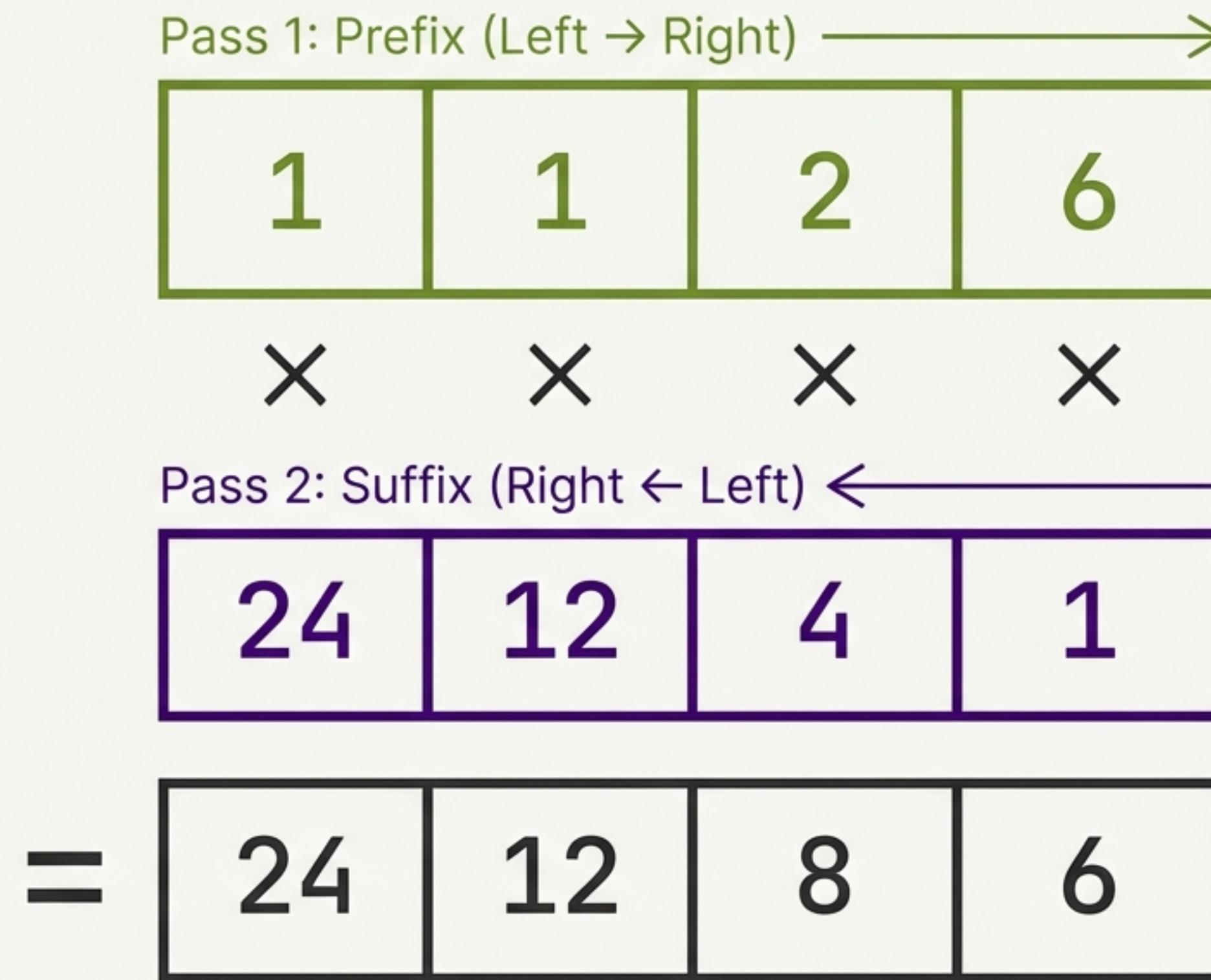


The Insight: Left × Right

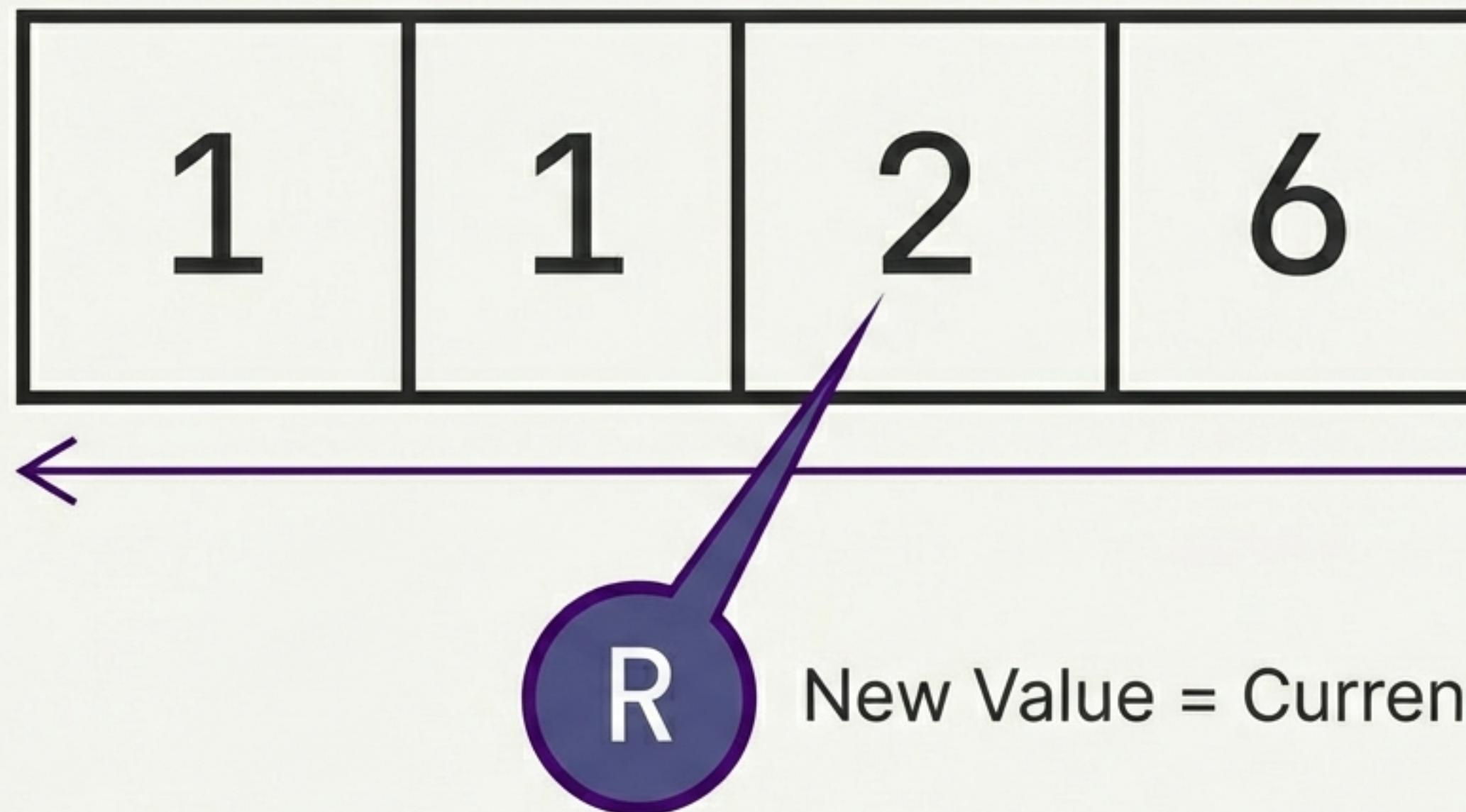


Result[i] = (Left Product) × (Right Product)

Two-Pass Construction



Space Optimization to O(1)



The Solution & Takeaway

```
def productExceptSelf(nums):
    res = [1] * len(nums)
    prefix = 1
    for i in range(len(nums)):
        res[i] = prefix
        prefix *= nums[i]
    postfix = 1
    for i in range(len(nums) - 1, -1, -1):
        res[i] *= postfix
        postfix *= nums[i]
    return res
```

Big O Analysis

- **Time:** $O(n)$
(Two linear passes)
- **Space:** $O(1)$
(Excluding output array)

The Pattern

Pre-computation. Break the problem into sub-problems (Prefix/Suffix) and merge.