Product of array except self.

example: 1,2,3,4 => 24, 12, 8, 6

0 1 2 3

Solution analytics O(n)

24 = 2×3×4 1×2×3×4 = 24

12 = 1×3×4

8 = 1×2×4

6 = 1×2×3

8 1 2 4

output[2] = nums[0] + nums[1] × nums[3]

2

×

output i: the product of right products & and left products

0 1 2 3

1, 2, 3, 4

left products of : i

0 → 1

1 → 1 × 1 = 1

2 → 2 × 1 = 2

3 → 3 × 2 = 6

end.

num[i-1] * left[i-1]

1, 1, 2, 6

nums[i+1] * right[i+1]

right products of : i

3 → 1

2 → 4 × 1 = 4

1 → 3 × 4 = 12

0 → 2 × 12 = 24

end

24, 12, 4, 1

output[0]: left[0] * right[0] = 24×1 = 24

output[1]: left[1] * right[1] = 1×12 = 12

output[2]: left[2] * right[2] = 2×4 = 8

3 3 3 = 6×1 = 6

```
class Solution {
public:
    vector<int> productExceptSelf(vector<int> &nums) {
        vector<int> left(nums.size());
        left[0] = 1;

        vector<int> right(nums.size());
        right[nums.size() - 1] = 1;

        for (int i = 1; i < nums.size(); i++) {
            left[i] = nums[i - 1] * left[i - 1];
        }

        for (int i = nums.size() - 2; i >= 0; i--) {
            right[i] = nums[i + 1] * right[i + 1];
        }

        vector<int> output(nums.size());
        for (int i = 0; i < nums.size(); i++) {
            output[i] = left[i] * right[i];
        }

        return output;
    }
};
```

we can remove the third loop and make its logic inside the second one .