

一、题目说明

题目238. Product of Array Except Self, 给n个整数, 返回一个数组, 每个元素都是原数组除了该位置元素外的乘积。不能用除法, 复杂度要求是O(n)。难度是Medium!

二、我的解答

看到这个问题, 我第一想法就是求所有元素的乘积, 然后就简单了。但是不允许使用除法...

然后能想到的思路就是可以用动态规划: $dp[i] = \text{nums}[0] * \dots * \text{nums}[i-1] = dp[i-1] * \text{nums}[i-1]$

可以将result[]作为dp使用, 技巧在于, 从左到右, 从右到左。

```
class Solution{
public:
    //dp, 其中dp[i]=nums[0]*...*nums[i-1] = dp[i-1]*nums[i-1]
    vector<int> productExceptSelf(vector<int>& nums){
        vector<int> result(nums.size());
        int k = 1;
        for(int i=0;i<result.size();i++){
            result[i] = k;
            k = k* nums[i];
        }
        k = 1;
        for(int i =result.size()-1;i>=0;i--){
            result[i] *= k;
            k *= nums[i];
        }
        return result;
    }
};
```

性能如下:

```
Runtime: 40 ms, faster than 85.10% of C++ online submissions for Product of
Array Except Self.
Memory Usage: 12.5 MB, less than 96.97% of C++ online submissions for Product of
Array Except Self.
```

三、优化措施

无