一、题目说明

题目152. Maximum Product Subarray，给一列整数，求最大连续子序列，其乘积最大。难度是Medium！

二、我的解答

这个题目，用双重循环就可以了。

```
class Solution{
    public:
        int maxProduct(vector<int>& nums){
            if(nums.size()<=1) return nums[0];
            product = INT_MIN;
            int len = nums.size();

            for(int i=0;i<len;i++){
                int p = nums[i];
                if(p>product) product = p;
                for(int t=i+1;t<len;t++){
                    p *= nums[t];
                    if(p>product) product = p;
                }
            }
            return product;
        }
    private:
        int product;
};
```

性能如下：

```
Runtime: 200 ms, faster than 6.16% of C++ online submissions for Maximum Product
Subarray.
Memory Usage: 9.1 MB, less than 82.50% of C++ online submissions for Maximum
Product Subarray.
```

三、优化措施

仔细再读读题目，一列整数，上述方法太"通用"，一次循环就可以了。

```
class Solution{
    public:
        //dp，其中dp[i]表示以第i个元素结尾的最大乘积
        int maxProduct(vector<int>& nums){
            if(nums.size()<=1) return nums[0];
            dpMax = nums[0];
            dpMin = nums[0];
            maxProd = nums[0];
            int len = nums.size();

            for(int i=1;i<len;i++){
                int preMax = dpMax;
                dpMax = max(dpMin*nums[i],max(nums[i],dpMax*nums[i]));
                dpMin = min(dpMin*nums[i],min(preMax*nums[i],nums[i]));
                cout<<"i="<<i<<",dpMax="<<dpMax<<",dpMin="<<dpMin<<"\n";
                maxProd = max(dpMax,maxProd);
            }
            return maxProd;
        }
    private:
        int dpMax;
        int dpMin;
        int maxProd;
};
```

```
Runtime: 16 ms, faster than 9.45% of C++ online submissions for Maximum Product
Subarray.
Memory Usage: 9.1 MB, less than 75.00% of C++ online submissions for Maximum
Product Subarray.
```