

## 连续子数组的最大和

输入一个整型数组，数组中的一个或连续多个整数组成一个子数组。求所有子数组的和的最大值。

要求时间复杂度为  $O(n)$ 。

示例 1:

输入: `nums = [-2,1,-3,4,-1,2,1,-5,4]`

输出: 6

解释: 连续子数组 `[4,-1,2,1]` 的和最大, 为 6。

解题思路:

动态规划

$Dp[i]$  表示 `nums` 中以 `nums[i]` 结尾的最大子数组和

## 动态规划法

数组:	2	3	-6	2	4
dp:	2	5	-1	2	6

$dp[i]$  表示  
`nums` 中以 `nums[i]` 结尾的最大子序和  
 $dp[0] = nums[0]$   
 $dp[i] = \max(dp[i-1] + nums[i], nums[i]);$

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        if(nums.size()==0) return 0;
        if(nums.size()==1) return nums[0];
        vector<int> dp(nums.size(),0);
        dp[0]=nums[0];
        int result=nums[0];
        for(int i=1;i<nums.size();i++)
        {
            dp[i]=dp[i-1]>0? (nums[i]+dp[i-1]):nums[i]; //非负就添加
            if(dp[i]>result) result=dp[i]; //更新最大值结果
        }
        return result;
    }
};
```