

剑指offer42.连续子数组的最大和

题目描述

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。

来源：力扣（LeetCode）

链接：<https://leetcode-cn.com/problems/lian-xu-zi-shu-zu-de-zui-da-he-lcof>

解题思路

- 动态规划(dynamic programming)，题目中给定了一个长度为 n 的数组`nums`,
- 定义状态：
 - $f(i)$ ——>表示以`nums[i]`结尾的连续子数组的最大和
- 状态转移方程， $f(i)$ 表示以`nums[i]`结尾的连续子数组最大和， $f(i)$ 应该等于 $f(i-1)$ 加上`nums[i]`，但是如果 $f(i-1)$ 为负数相加之后的结果反倒小于`nums[i]`，所以我们应该判断一个`nums[i]`和 $f(i-1)+\text{nums}[i]$ 之间的大小，
 - $f(i)=\max(f(i-1)+\text{nums}[i], \text{nums}[i])$
- 初始化：
 - `ret=nums[0]`
- 返回结果，定义一个临时变量，一直更新存储连续子数组和的最大值
 - `max(f(i))`

代码实现

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int ret=nums[0];
        for(int i=1;i<nums.size();++i)
        {
            nums[i]=max(nums[i-1]+nums[i],nums[i]);
            ret=max(ret,nums[i]);
        }
        return ret;
    }
};
```