

```
import Foundation
/// 剑指 Offer 42. 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)。
class Solution {
    /// 执行用时：96 ms, 在所有 Swift 提交中击败了 97.87% 的用户
    /// 内存消耗：16.2 MB, 在所有 Swift 提交中击败了 27.66% 的用户
    /// 通过测试用例：202 / 202
    func maxSubArray(_ nums: [Int]) -> Int {
        var _nums = nums
        for i in 1..<_nums.count {
            _nums[i] += max(_nums[i-1], 0)
        }
        return _nums.max() ?? 0
    }
}
let solution = Solution()
print(solution.maxSubArray([-2,1,-3,4,-1,2,1,-5,4]))
```