Maximum Subarray (/problems/maximum-subarray/)

Submission Detail

200 / 202 test cases passed. Status: Time Limit Exceeded

Submitted: **5 minutes ago**

Last executed input: `[-57,9,-72,-72,-62,45,-97,24,-39,35,-82,-4,-63,1,-93,42,44,1,-75,-25,-87,-16,9,-59,20,5,-95,-41,4,-30,4`

Submitted Code: 5 minutes ago

Language: swift

Edit Co

```swift
func myPrint(_ arg: Any) {
    //print(arg)
}

class Solution {
    func maxSubArray(_ nums: [Int]) -> Int {
        assert(nums.count > 0)

        return maxSubArrayA(nums)
    }

    // Time: O(nlogn)
    // Space: O(n)
    private func maxSubArrayA(_ nums: [Int]) -> Int {
        var maxSum = nums[0]
        myPrint(maxSum)
        var i = 0
        while i < nums.count {
            var j = i + 1
            var newSum = nums[i]
            if newSum > maxSum {
                maxSum = newSum
                myPrint("update max to \(maxSum)")
            }
            myPrint("i: \(i), j: \(j), newSum: \(newSum)")
            while j < nums.count {
                newSum += nums[j]
                myPrint("i: \(i), j: \(j), newSum: \(newSum)")
                if newSum > maxSum {
                    maxSum = newSum
                    myPrint("update max to \(maxSum)")
                }
                j += 1
            }
            myPrint("")
            i += 1
        }
        return maxSum
    }
}
```

Back to problem (/problems/maximum-subarray/)

Copyright © 2020 LeetCode

States (/re