

```

import Foundation
/// 剑指 Offer 30. 包含min函数的栈
/// 定义栈的数据结构，请在该类型中实现一个能够得到栈的
/// 最小元素的 min 函数在该栈中，调用 min、push 及 pop 的时间复杂度都是 O(1)。
class MinStack {
    var stack: [Int] = []
    /// 辅助Stack，用来存取每次读取的最小值
    /// 在获取最小值的时候直接返回`assistStack`的最后一个元素即可
    var assistStack: [Int] = []

    init() {}

    func push(_ x: Int) {
        stack.append(x)
        if assistStack.isEmpty {
            assistStack.append(x)
        } else {
            if let last = assistStack.last {
                if last >= x {
                    assistStack.append(x)
                }
            }
        }
    }

    func pop() {
        if let last1 = stack.last {
            if let last2 = assistStack.last {
                if last1 == last2 {
                    assistStack.popLast()
                }
            }
        }
        stack.popLast()
    }

    func top() -> Int {
        if let last = stack.last {
            return last
        }
        return -1
    }

    func min() -> Int {
        if let last = assistStack.last {
            return last
        }
        return -1
    }
}

```