

```

import Foundation
/// 剑指 Offer 59 - II. 队列的最大值
/// 请定义一个队列并实现函数 max_value 得到队列里的最大值，
/// 要求函数max_value、push_back 和 pop_front 的均摊时间复杂度都是 $O(1)$ 。
///
/// 若队列为空，pop_front 和 max_value 需要返回 -1
class MaxQueue {

    private var queue: [Int] = []
    /// 辅助双向队列(可以insert也可以append), 递减序列. 最大值永远在第一个数
    private var assistQueue: [Int] = []

    init() {}

    func max_value() -> Int {
        if let first = assistQueue.first { return first }
        return -1
    }

    func push_back(_ value: Int) {
        /// queue直接在第一位插入
        queue.insert(value, at: 0)
        /// 辅助queue需要先删除所有比value小的数, 再插入新的数
        if assistQueue.isEmpty { assistQueue.insert(value, at: 0) }
        else {
            /// 辅助queue需要先删除所有比value小的数
            while (!assistQueue.isEmpty) {
                if let last = assistQueue.last {
                    if last < value { assistQueue.popLast() }
                    else { break }
                }
            }
            /// 再插入新的数
            assistQueue.append(value)
        }
    }

    func pop_front() -> Int {
        if queue.isEmpty { return -1 }
        if let last = queue.last {
            if let first = assistQueue.first {
                if last == first { assistQueue.remove(at: 0) }
            }
            queue.popLast()
            return last
        }
        return -1
    }
}

```