

```

import Foundation
/// **剑指 Offer 66. 构建乘积数组**
/// 给定一个数组 A[0,1,...,n-1]，请构建一个数组 B[0,1,...,n-1]，
/// 其中 B[i] 的值是数组 A 中除了下标 i 以外的元素的积，
/// 即  $B[i]=A[0]\times A[1]\times \dots \times A[i-1]\times A[i+1]\times \dots \times A[n-1]$ 。不能使用除法。
///
/// 示例：
/// 输入：[1, 2, 3, 4, 5]
/// 输出：[120, 60, 40, 30, 24]
class Solution {
    /// **核心思想：利用索引左侧所有数字的乘积和右侧所有数字的乘积相乘得到答案**
    /// 将数据进行左右两次遍历设数据为[4,5,6,8,2]
    ///     - 数据进行从左往右遍历的时候进行依次累乘同时注意忽略第一个数
    ///         leftRes = [1, 1x4, 1x4x5, 1x4x5x6, 1x4x5x6x8, 1x4x5x6x8x2]
    ///     - 在数据进行从右往左遍历的时候进行依次累乘同时注意忽略最后一个数
    ///         rightRes = [1x2x8x6x5, 1x2x8x6, 1x2x8, 1x2, 1]
    /// 最后两个数组内的数对应相乘即可得到最终的结果。
    func constructArr(_ a: [Int]) -> [Int] {
        if a.isEmpty { return [] }
        var res: [Int] = []
        /// 从左到右进行遍历
        var leftRes: [Int] = Array(repeating: 1, count: a.count)
        for i in 1..

```