

```

import Foundation
/// **剑指 Offer 51. 数组中的逆序对**
/// 在数组中的两个数字，如果前面一个数字大于后面的数字，
/// 则这两个数字组成一个逆序对。输入一个数组，求出这个数组中的逆序对的总数。
///
/// 示例 1:
/// 输入: [7, 5, 6, 4]
/// 输出: 5
class Solution {
    func reversePairs(_ nums: [Int]) -> Int {
        var numsCopy = nums
        // 返回逆序对的数量
        return mergeSort(&numsCopy, 0, nums.count - 1)
    }

    // 归并排序函数
    func mergeSort(_ nums: inout [Int], _ l: Int, _ r: Int) -> Int {
        // 终止条件
        if l >= r { return 0 }
        // 递归划分
        let m = (l + r) / 2
        var res = mergeSort(&nums, l, m) + mergeSort(&nums, m + 1, r)
        // 合并阶段
        var i = l, j = m + 1
        // 用于合并阶段的临时数组
        var merged = [Int]()
        while i <= m || j <= r {
            if j > r || (i <= m && nums[i] <= nums[j]) {
                merged.append(nums[i])
                i += 1
            } else {
                merged.append(nums[j])
                j += 1
                // 统计逆序对
                res += m - i + 1
            }
        }
        // 将合并后的数组拷贝回原数组 nums
        for (index, value) in merged.enumerated() {
            nums[l + index] = value
        }
        return res
    }
}

let solution = Solution()
print(solution.reversePairs([7,5,6,4]))

```