

```

import Foundation
// 剑指 Offer 58 - I. 翻转单词顺序
// 输入一个英文句子，翻转句子中单词的顺序，但单词内字符的顺序不变。
// 为简单起见，标点符号和普通字母一样处理。
// 例如输入字符串"I am a student. "，则输出"student. a am I"。
class Solution {
    /// 执行用时：36 ms, 在所有 Swift 提交中击败了 100.00% 的用户
    /// 内存消耗：14.3 MB, 在所有 Swift 提交中击败了 87.50% 的用户
    /// 通过测试用例：25 / 25
    /// 时间复杂度：O(n)
    /// 空间复杂度：O(1)
    func reverseWordsSolution1(_ s: String) -> String {
        return String(s.split(separator: " ").reversed().joined(separator: " "))
    }

    /// 双指针超出时间限制，嵌套遍历和获取 `String.index` 对象会使用更多时间
    func reverseWordsSolution2(_ s: String) -> String {
        /// 先删除首尾的空格
        let s_ = s.trimmingCharacters(in: .whitespaces)

        var result: [String] = []
        var i = s_.count - 1
        var j = i
        let startIndex = s_.startIndex

        while (i >= 0) {
            while (i >= 0) && (s_[s_.index(startIndex, offsetBy: i)] != " ") {
                i -= 1
            }
            let temp = s_[s_.index(startIndex, offsetBy: i+1)...s_.index(startIndex, offsetBy: j)]
            result.append(String(temp))
            while (i >= 0) && (s_[s_.index(startIndex, offsetBy: i)] == " ") {
                i -= 1
            }
            j = i
        }

        return String(result.joined(separator: " "))
    }

    /// 执行用时：44 ms, 在所有 Swift 提交中击败了 67.5% 的用户
    /// 内存消耗：14.3 MB, 在所有 Swift 提交中击败了 87.50% 的用户
    /// 通过测试用例：25 / 25
    func reverseWords(_ s: String) -> String {
        var result: [String] = []

        var temp = ""

```

```

for c in s {
    if c != " " {
        temp += String(c)
    } else {
        /// 通过判断是否已经存放字符来决定是否重制`temp`
        if temp.count > 0 {
            result.insert(temp, at: 0)
            temp = ""
        }
    }
}
/// 防止最后没有空格导致的缺少单词的情况
if temp.count > 0 {
    result.insert(temp, at: 0)
}
return String(result.joined(separator: " "))
}

```