

```

import Foundation
/// **剑指 Offer 38. 字符串的排列解法 2**
/// 输入一个字符串，打印出该字符串中字符的所有排列。
/// 你可以以任意顺序返回这个字符串数组，但里面不能有重复元素。
///
/// 示例:
/// 输入: s = "abc"
/// 输出: ["abc", "acb", "bac", "bca", "cab", "cba"]
///
/// 回溯算法的介绍可参考: https://mp.weixin.qq.com/s/nMUHqvWzG2LmWA9jMIHwQQ
class Solution {
    private var res: Set<String> = []
    private var sArray: [String] = []

    func permutation(_ s: String) -> [String] {
        /// LeetCode 没有办法直接使用 s.split(separator: "")
        s.forEach { c in self.sArray.append(String(c)) }
        /// 用于表示字符是否已经被使用
        var visited: [Bool] = Array(repeating: false, count: s.count)
        var track: [String] = []
        backtrack(&track, sArray, &visited)
        return Array(res)
    }

    /// 回溯算法
    /// - Parameter track: **路径**，当前固定位，index之前表示已经做出的选择
    /// - Parameter sArray: **选择列表**
    /// - Parameter visited: 字符是否被使用过
    private func backtrack(_ track: inout [String],
                           _ sArray: [String],
                           _ visited: inout [Bool]) {
        if track.count == sArray.count {
            res.insert(track.reduce("", +))
            return
        }
        for id in 0..

```