```
import Foundation
/// **剑指 Offer 38. 字符串的排列解法 1**
/// 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: [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)) }
       dfs(index: 0)
       return res
    }
    /// 回溯算法
    /// - Parameter index: **路径**, 当前固定位, index之前表示已经做出的选择
   private func dfs(index: Int) {
       /// 回溯算法终止的条件, 代表所有位已固定
       if index == self.sArray.count - 1 {
           self.res.append(sArray.reduce("", +))
           return
       }
       var letters: Set<String> = []
       for i in index..<self.sArray.count {</pre>
           /// 做出选择
           if letters.contains(self.sArray[i]) { continue }
           letters.insert(self.sArray[i])
           sArray.swapAt(i, index)
           dfs(index: index + 1)
           /// 撤销选择
           sArray.swapAt(i, index)
       }
   }
}
let solution = Solution()
print(solution.permutation("abc"))
```