

```

import Foundation
/// 剑指 Offer 46. 把数字翻译成字符串
/// 给定一个数字，我们按照如下规则把它翻译为字符串：
/// 0 翻译成“a”，1 翻译成“b”，……，11 翻译成“l”，……，25 翻译成“z”。
/// 一个数字可能有多个翻译。请编程实现一个函数，用来计算一个数字有多少种不同的翻译方法。
///
/// 示例 1:
/// 输入: 12258
/// 输出: 5
/// 解释: 12258有5种不同的翻译，分别是"bccfi", "bwfi", "bczi", "mcfi"和"mzi"
class Solution {
    func translateNum(_ num: Int) -> Int {
        if String(num).count == 1 { return 1 }

        /// `F(0) = F(1) = 1`
        var (a, b): (Int, Int) = (1, 1)
        /// 先将 num 转为字符串并计算 num 数字位数
        let (numStr, length): (String, Int) = (String(num), String(num).count)
        let startIndex = numStr.startIndex
        for id in 2...length {
            /// 比较直观的写法，但不是很推荐比较浪费内存
            let slice = String(Array(numStr)[id-2..

```