

```

import Foundation
/// 剑指 Offer 03. 数组中重复的数字
/// 找出数组中重复的数字。
/// 在一个长度为 n 的数组 nums 里的所有数字都在 0~n-1 的范围内。
/// 数组中某些数字是重复的，但不知道有几个数字重复了，也不知道
/// 每个数字重复了几次。请找出数组中任意一个重复的数字。
///
/// 示例 1:
/// 输入: [2, 3, 1, 0, 2, 5, 3]
/// 输出: 2 或 3
class Solution {
    /// 执行用时: 224 ms, 在所有 Swift 提交中击败了 97.85% 的用户
    /// 内存消耗: 17.8 MB, 在所有 Swift 提交中击败了 39.78% 的用户
    /// 通过测试用例: 27 / 27
    func findRepeatNumberSolution1(_ nums: [Int]) -> Int {
        var set: [Int: Int] = [:]
        for num in nums {
            if let value = set[num] {
                return num
            } else {
                set[num] = 0
            }
        }
        return -1
    }

    func findRepeatNumber(_ nums: [Int]) -> Int {
        var nums_ = nums
        var id = 0
        var length = nums_.count
        while id < length {
            /// 当前的数字与位置是相同对应的则不做操作
            if nums_[id] == id {
                id += 1
                continue
            }
            /// 在别处找到相同的数字，则说明是重复数字
            if nums_[nums_[id]] == nums_[id] {
                return nums_[id]
            }
            (nums_[nums_[id]], nums_[id]) = (nums_[id], nums_[nums_[id]])
            print(nums_)
        }
        return -1
    }
}

let solution = Solution()
print(solution.findRepeatNumber([2, 3, 1, 0, 2, 5, 3]))

```