```swift
import Foundation
// 剑指 Offer 20. 表示数值的字符串
// 请实现一个函数用来判断字符串是否表示数值（包括整数和小数）。
//
// 数值（按顺序）可以分成以下几个部分：
// 1. 若干空格
// 2. 一个 小数 或者 整数
// 3. （可选）一个 'e' 或 'E' ，后面跟着一个 整数
// 4. 若干空格
//
// 小数（按顺序）可以分成以下几个部分：
// 1. （可选）一个符号字符（'+' 或 '-'）
// 2. 下述格式之一：
//       至少一位数字，后面跟着一个点 '.'
//       至少一位数字，后面跟着一个点 '.' ，后面再跟着至少一位数字
//       一个点 '.' ，后面跟着至少一位数字
//
// 整数（按顺序）可以分成以下几个部分：
// 1. （可选）一个符号字符（'+' 或 '-'）
// 2. 至少一位数字
//
// 部分数值列举如下：
// ["+100", "5e2", "-123", "3.1416", "-1E-16", "0123"]
//
// 部分非数值列举如下：
// ["12e", "1a3.14", "1.2.3", "+-5", "12e+5.4"]
class Solution {
    private let states: [[Character: Int]] = [
        /// 0. start with 'blank'
        [" ": 0, "s": 1, "d": 2, ".": 4],
        /// 1. 'sign' before 'e'
        ["d": 2, ".": 4 ],
        /// 2. 'digit' before 'dot'
        ["d": 2, ".": 3, "e": 5, " ": 8],
        /// 3. 'digit' after 'dot'
        ["d": 3, "e": 5, " ": 8],
        /// 4. 'digit' after 'dot' (‘blank’ before 'dot')
        ["d": 3],
        /// 5. 'e'
        ["s": 6, "d": 7],
        /// 6. 'sign' after 'e'
        ["d": 7],
        /// 7. 'digit' after 'e'
        ["d": 7, " ": 8],
        /// 8. end with 'blank'
        [" ": 8]
    ]
```

```
    func isNumberSolution1(_ s: String) -> Bool {
        /// Start with state 0
        var current_state: Int = 0
        for c in s {
            var t: Character = Character(" ")
            if c.isNumber { t = "d" }
            else if "+-".contains(where: { $0 == c }) { t = "s" }
            else if "eE".contains(where: { $0 == c }) { t = "e" }
            else if ". ".contains(where: { $0 == c }) { t = c }
            else { t = "?" }
            if let state_num = states[current_state][t] {
                current_state = state_num
            } else {
                return false
            }
        }
        return [2,3,7,8].contains(where: { $0 == current_state })
    }

    func isNumber(_ s: String) -> Bool {
        return s.range(of: #"^
         *([-\+]?\d+((\.\d+)?|\.?)|([-\+]?\.\d+))([Ee][\+-]?\d+)? *$"#,
                       options: .regularExpression) != nil
    }
}
```