

```

import Foundation
// 剑指 Offer 04. 二维数组中的查找
// 在一个 n * m 的二维数组中，每一行都按照从左到右非递减的顺序排序
// 每一列都按照从上到下非递减的顺序排序。
// 请完成一个高效的函数，输入这样的一个二维数组和一个整数，判断数组中是否含有该整数。
//
// 示例：
//
// 现有矩阵 matrix 如下：
// [
//     [1,   4,   7,  11, 15],
//     [2,   5,   8,  12, 19],
//     [3,   6,   9,  16, 22],
//     [10, 13, 14, 17, 24],
//     [18, 21, 23, 26, 30]
// ]
// 给定 target = 5, 返回 true。
// 给定 target = 20, 返回 false。
class Solution {
    func findNumberIn2DArraySolution1(_ matrix: [[Int]], _ target: Int) ->
        Bool {
        if matrix.isEmpty { return false }
        var length: Int = matrix[0].count
        for row in matrix {
            /// 每一行按照二分法进行查找
            var left: Int = 0
            var right: Int = length-1
            var id = 0
            while (left <= right) {
                var mid = left + (right-left)/2
                if row[mid] == target { return true }
                else if row[mid] < target { left = mid + 1 }
                else { right = mid - 1 }
            }
        }
        return false
    }

    func findNumberIn2DArray(_ matrix: [[Int]], _ target: Int) -> Bool {
        var (i, j): (Int, Int) = (matrix.count - 1, 0)
        while (i >= 0) && (j < matrix[0].count) {
            if matrix[i][j] > target { i -= 1 }
            else if matrix[i][j] < target { j += 1 }
            else { return true }
        }
        return false
    }
}

```