

```

import Foundation
/// 剑指 Offer 11. 旋转数组的最小数字
/// 把一个数组最开始的若干个元素搬到数组的末尾，我们称之为数组的旋转。
///
/// 给你一个可能存在 重复 元素值的数组 numbers，它原来是一个升序排列的数组，
/// 并按上述情形进行了一次旋转。请返回旋转数组的最小元素。
/// 例如，数组 [3,4,5,1,2] 为 [1,2,3,4,5] 的一次旋转，该数组的最小值为 1。
///
/// 注意，数组 [a[0], a[1], a[2], ..., a[n-1]] 旋转一次 的结果为数组 [a[n-1], a[0], a[1], a[2], ...,
    a[n-2]]。
class Solution {
    func minArray(_ numbers: [Int]) -> Int {
        var left = 0
        var right = numbers.count - 1
        while (left < right) {
            let mid = left + (right - left) >> 2
            if numbers[mid] < numbers[right] {
                /// 如果当前的数小于最右边的数，则表明断裂点必然在左边
                /// 这道题与其他题目不太一样，这里不是`right = mid - 1`
                /// 因为要找最小的 nums[m]有可能是最小的。
                /// 比如[4,5,1,2,3] 所以`j=m`
                right = mid
            } else if numbers[mid] > numbers[right] {
                /// 如果当前的数大于最右边的数，则表明断裂点必然在右边
                left = mid + 1
            } else {
                /// 去除重复的元素
                right -= 1
            }
        }
        return numbers[left]
    }
}

let s = Solution()
print(s.minArray([3,4,5,1,2]))
print(s.minArray([2,2,2,0,1]))
print(s.minArray([3,1,3]))
print(s.minArray([3,1,3,3,3]))

```