```swift
import Foundation
/// **剑指 Offer 41. 数据流中的中位数**
/// 如何得到一个数据流中的中位数？如果从数据流中读出奇数个数值，
/// 那么中位数就是所有数值排序之后位于中间的数值。如果从数据流中读出偶数个数值，
/// 那么中位数就是所有数值排序之后中间两个数的平均值。
/// 例如，
/// [2,3,4] 的中位数是 3
/// [2,3] 的中位数是 (2 + 3) / 2 = 2.5
/// 设计一个支持以下两种操作的数据结构：
/// `void addNum(int num)` - 从数据流中添加一个整数到数据结构中。
/// `double findMedian()` - 返回目前所有元素的中位数。
/// 示例 1：
/// 输入：
/// ["MedianFinder", "addNum", "addNum", "findMedian", "addNum", "findMedian"]
/// `[[],[1],[2],[],[3],[]]`
/// 输出：[null, null, null, 1.50000, null, 2.00000]
class MedianFinder {
    private var arr: [Int] = [] // 数组用于按排序顺序存储数据流
    init() {}

    func addNum(_ num: Int) {
        // 如果数组为空，则直接将数字添加到数组中
        if arr.count == 0 { arr.append(num) }
        else {
            // 使用二分搜索找到正确的插入位置，以保持排序顺序
            var (low, high) = (0, arr.count - 1)
            while low <= high {
                var mid = (high + low) / 2
                if num >= arr[mid] { low = mid + 1 }
                else { high = mid - 1 }
            }
            arr.insert(num, at: low)
        }
    }

    func findMedian() -> Double {
        var count = arr.count
        // 如果数组元素个数为偶数，则计算中位数为中间两个元素的平均值
        if count % 2 == 0 {
            let mid = count / 2 - 1
            let next = mid + 1
            return Double(arr[mid] + arr[next]) / 2.0
        } else {
            // 如果数组元素个数为奇数，则中位数为中间元素
            return Double(arr[count / 2])
        }
    }
}
```