# 41.1 数据流中的中位数

## 题目链接

[牛客网](https://www.nowcoder.com/practice/9be0172896bd43948f8a32fb954e1be1?tpId=13&tqId=11216&tPage=1&rp=1&ru=/ta/coding-interviews&qru=/ta/coding-interviews/question-ranking&from=cyc_github)

## 题目描述

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。

## 解题思路

/\* 大顶堆，存储左半边元素 \*/
private PriorityQueue<Integer> left = new PriorityQueue<>((o1, o2) -> o2 - o1);
/\* 小顶堆，存储右半边元素，并且右半边元素都大于左半边 \*/
private PriorityQueue<Integer> right = new PriorityQueue<>();
/\* 当前数据流读入的元素个数 \*/
private int N = 0;

public void Insert(Integer val) {
 /\* 插入要保证两个堆存于平衡状态 \*/
 if (N % 2 == 0) {
 /\* N 为偶数的情况下插入到右半边。
 \* 因为右半边元素都要大于左半边，但是新插入的元素不一定比左半边元素来的大，
 \* 因此需要先将元素插入左半边，然后利用左半边为大顶堆的特点，取出堆顶元素即为最大元素，此时插入右半边 \*/
 left.add(val);
 right.add(left.poll());
 } else {
 right.add(val);
 left.add(right.poll());
 }
 N++;
}

public Double GetMedian() {
 if (N % 2 == 0)
 return (left.peek() + right.peek()) / 2.0;
 else
 return (double) right.peek();
}