链滴

数据流中的中位数

作者：yudake

原文链接：https://ld246.com/article/1519089363407

来源网站：链滴

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均值。

解题思路

需要两个数据结构，能够对进入数据排序，并且两数据结构大小小于等于1。中位数只在数据交接点现。

采用PriorityQueue实现大顶堆和小顶堆。

Java默认是小顶堆，重新定义Comparator实现大顶堆。

同时满足以下条件：

- 两个堆中的数据数目差不能超过1，这样可以使中位数只会出现在两个堆的交接处；
- 大顶堆的所有数据都小于小顶堆，这样就满足了排序要求。

代码

```
import java.util.Comparator;
import java.util.PriorityQueue;
public class Solution {
    int count;
    PriorityQueue<Integer> minHeap = new PriorityQueue<Integer>();
    PriorityQueue<Integer> maxHeap = new PriorityQueue<Integer>(11, new Comparator<Int
ger>() {
        @Override
        public int compare(Integer o1, Integer o2) {
            //PriorityQueue默认是小顶堆，实现大顶堆，需要反转默认排序器
            return o2.compareTo(o1);
        }
    });

    public void Insert(Integer num) {
        count++;
        if ((count & 1) == 0) {
            if (!maxHeap.isEmpty() && num < maxHeap.peek()) {
                maxHeap.offer(num);
                num = maxHeap.poll();
            }
            minHeap.offer(num);
        } else {
            if (!minHeap.isEmpty() && num > minHeap.peek()) {
                minHeap.offer(num);
                num = minHeap.poll();
            }
            maxHeap.offer(num);
        }
```

```
    }

    public Double GetMedian() {
        if (count == 0)
            return 0.0;
        if ((count & 1) == 1)
            return (double)maxHeap.peek();
        else
            return (double)(minHeap.peek() + maxHeap.peek()) / 2.0;
    }
}
```

原文链接：数据流中的中位数