一、题目说明

题目295. Find Median from Data Stream，数据流的中位数（数据为奇数个，则为中间的；否则为中间2个数的平均数）。

二、我的解答

用一个数组实现，但是超时：

```
class MedianFinder {
    vector<int> store;

public:
    void addNum(int num)
    {
        if (store.empty())
            store.push_back(num);
        else
            store.insert(lower_bound(store.begin(), store.end(), num), num);
    }
    double findMedian()
    {
        int n = store.size();
        return n & 1 ? store[n / 2] : (store[n / 2 - 1] + store[n / 2]) * 0.5;
    }
};
```

```
Runtime: 304 ms, faster than 9.04% of C++ online submissions for Find Median
from Data Stream.
Memory Usage: 42.7 MB, less than 60.87% of C++ online submissions for Find
Median from Data Stream.
```

三、优化措施

用2个优先级队列实现：

```
class MedianFinder {
    priority_queue<int> max;//大根堆
    priority_queue<int, vector<int>, greater<int>> min;
public:
    //数据优先放入大顶堆中，然后将大顶堆堆顶元素放入小顶堆中
    //如果大顶堆元素数量小于小顶堆元素数量，则从小顶堆中弹出一个元素放入大顶堆中
    //如果数据总量为奇数，则中位数为大顶堆堆顶元素，否则为大顶堆和小顶堆元素和除以2。
    MedianFinder() {

    }

    void addNum(int num) {
        max.push(num);
        min.push(max.top());
        max.pop();
        if (max.size() < min.size()) {
            max.push(min.top());
            min.pop();
        }
    }

    double findMedian() {
        int size = max.size() + min.size();
        return size % 2 == 1 ? max.top() : (max.top() + min.top()) / 2.0;
    }
};
```

```
Runtime: 172 ms, faster than 47.48% of C++ online submissions for Find Median
from Data Stream.
Memory Usage: 42.4 MB, less than 100.00% of C++ online submissions for Find
Median from Data Stream.
```