面试题 41：数据流中的中位数

题目：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：堆

1. 最大堆实现左半部分数据，最小堆实现右半部分数据。插入一个数O(logn)，得到中位数O(1)
2. 平均分配，两个堆中数据数目之差不能超过1。总数是偶数时，插入最小堆；总数是奇数时，插入最大堆
3. 要保证最大堆中数据都小于最小堆的数据。所以，总数是偶数时，先插入最大堆，然后把最大堆中的最大数插入最小堆；奇数时同理

3、pop_back/push_back，push_heap/pop_heap的区别

3.1 back系列和heap系列的区别：

back系列是操作容器元素添加和删除的方法，heap是堆排序的方法。不要混淆。

3.2 pop_back VS push_back

pop_back将一个容器的最后一个元素删除，push_back是往容器最后添加一个元素。 两者一般都没有返回值。

3.3 push_heap/pop_heap的区别

（1）push_heap是在容器添加一个元素后，将其堆排序，less（）对应最大堆排序，greater()对应最小堆排序。less和greater定义在头文件中。该方法一般用在向容器最后添加了元素之后。

（2）pop_heap是将当前容器的第一元素移到最后，并将剩余的元素堆排序。less（）对应最大堆排序，greater()对应最小堆排序，同之前。

```
vector<int> min;
vector<int> max;
void insert(int num)
{
    int size=min.size()+max.size();
    if((size&1)==0)
    {
        if(max.size()>0 && num<max[0])
        {
            max.push_back(num);
            push_heap(max.begin(),max.end(),less<int>());
            num=max[0];
            pop_heap(max.begin(),max.end(),less<int>());
            max.pop_back();
        }
        min.push_back(num);
        push_heap(min.begin(),min.end(),greater<int>());
    }
    else
    {
        if(min.size()>0 && num>min[0])
        {
            min.push_back(num);
            push_heap(min.begin(),min.end(),greater<int>());
            num=min[0];
            pop_heap(min.begin(),min.end(),greater<int>());
            min.pop_back();
        }
        max.push_back(num);
        push_heap(min.begin(),min.end(),less<int>());
    }
}
double getMedian()
{
    int size=min.size()+max.size();
    if(size<=0)
        return 0.0;
    if((size&1)==0)
        return (min[0]+max[0])/2.0;
    else
        return min[0];
}
```