Median 说明文档

一、项目说明：

Median 类的 public static int[] findMedians(int[] array)方法，实现了时间复杂度 O(NlogN)的带插入全局中间值在线查询，main 方法进行了 sample input.txt 的测试.

GenData 类会在 extra data 文件夹中随机生成 10 组测试数据，TestProg 类中的 main 方法会测试 testing case 文件夹中给出的 10 组测试数据和 extra data 文件夹中的 10 组测试数据.

二、算法逻辑及伪代码：

```
创建最大堆 maxHeap
创建最小堆 minHeap
for (int i = 0; i < array.length; i++) {
    if (maxHeap 的 size 为 0 || maxHeap 中的最大值大于 array[i]){
        将 array[i]加入 maxHeap
        if (maxHeap 的 size 大于 minHeap 的 size 加 1){
            将 maxHeap 中的最大元素弹出并加入 minHeap
        }
    } else {
        将 array[i]加入 minHeap
        if (maxHeap 的 size 小于 minHeap 的 size){
            将 minHeap 中的最小元素弹出并加入 maxHeap
        }
    }
    此时 maxHeap 中的最大值就是全局中间值，赋值给 array[i]
}
```

三、复杂度分析：

1、最好情况：$O(N)$

对于 array = [0, 0, 0, ... , 0] 或 array = [1, 2, 3, ... , N] 的情况，只需要 n 次插入和返回，不需要在 maxHeap 或 minHeap 中进行 swim()，因此最好情况的时间复杂度为 O(N)

2、最坏情况：$O(N\log N)$

在 maxHeap 或 minHeap 中插入一个元素，最坏情况是 log 级别，若每一次插入均为最坏情况，则总复杂度为 $\log 1+\log 2+...+\log N$，用积分近似计算发现，其结果为 $O(N\log N)$

$$\int_1^N \log x dx=\frac{1}{\ln 2}\int_1^N \ln x dx=\frac{1}{\ln 2}(x\ln x-x)|_1^N=N\log N-\frac{N}{\ln 2}+\frac{1}{\ln 2}=O(N\log N)$$

3、平均：$O(N\log N)$

设 T_n 表示第 n 个数插入时的平均比较和交换次数.

那么 $T_n=(n+1)+\left(\frac{T_0+T_{n-1}}{n}\right)+\left(\frac{T_1+T_{n-2}}{n}\right)+...+\left(\frac{T_{n-1}+T_0}{n}\right)$，$T_0=T_1=0$

化简得 $\frac{T_n}{n+1}=\frac{T_{n-1}}{n}+\frac{2}{n+1}$

$$T_N=2(N+1)\left(\frac{1}{3}+\frac{1}{4}+...+\frac{1}{N+1}\right)\approx 2(N+1)\int_3^{N+1}\frac{1}{x}dx\approx 1.39N\log N$$

所以，平均时间复杂度为 $O(N\log N)$