

```

        public int compare(Integer o1, Integer o2) {
            if (o2 > o1) {
                return 1;
            } else {
                return -1;
            }
        }
    }

    //生成小根堆的比较器
    public class MinHeapComparator implements Comparator<Integer> {
        @Override
        public int compare(Integer o1, Integer o2) {
            if (o2 < o1) {
                return 1;
            } else {
                return -1;
            }
        }
    }
}

```

## 在两个长度相等的排序数组中找到上中位数

### 【题目】

给定两个有序数组 `arr1` 和 `arr2`，已知两个数组的长度都为  $N$ ，求两个数组中所有数的上中位数。

### 【举例】

`arr1=[1,2,3,4]`，`arr2=[3,4,5,6]`

总共有 8 个数，那么上中位数是第 4 小的数，所以返回 3。

`arr1=[0,1,2]`，`arr2=[3,4,5]`

总共有 6 个数，那么上中位数是第 3 小的数，所以返回 2。

### 【要求】

时间复杂度为  $O(\log N)$ ，额外空间复杂度为  $O(1)$ 。

### 【难度】

尉 ★★★☆☆

## 【解答】

根据时间复杂度的要求可知，应该利用二分的方式寻找上中位数，具体过程为：

1. 重新定义一下问题，现在我们在  $\text{arr1}[\text{start1}..\text{end1}]$  与  $\text{arr2}[\text{start2}..\text{end2}]$  上寻找这两段数组共同的上中位数，并且这两段的长度应该相等( $\text{end1}-\text{start1}==\text{end2}-\text{start2}$ )。

2. 初始时  $\text{start1}=0$ ,  $\text{end1}=N-1$ ，即  $\text{arr1}[\text{start1}..\text{end1}]$  代表  $\text{arr1}$  的全部。 $\text{start2}=0$ ,  $\text{end2}=N-1$ ，即  $\text{arr2}[\text{start2}..\text{end2}]$  代表  $\text{arr2}$  的全部。

3. 如果  $\text{start1}==\text{end1}$ ，那么也有  $\text{start2}==\text{end2}$ ，找寻的过程中始终保证两段长度一致。这种情况下说明每一段都只有一个元素，这时元素总个数是 2 个，上中位数为较小的那个，则应该直接返回  $\min\{\text{arr1}[\text{start1}], \text{arr2}[\text{start2}]\}$ 。

4. 如果  $\text{start1}!=\text{end1}$ ，此时说明两段数组的长度都大于 1，则令  $\text{mid1}=(\text{start1}+\text{end1})/2$ ，代表  $\text{arr1}[\text{start1}..\text{end1}]$  的中间位置。令  $\text{mid2}=(\text{start2}+\text{end2})/2$ ，代表  $\text{arr2}[\text{start2}..\text{end2}]$  的中间位置。那么具体情况有三种。

情况一，如果  $\text{arr1}[\text{mid1}]==\text{arr2}[\text{mid2}]$ 。为了方便理解，举两个例子说明这种情况。

1)  $\text{arr1}$  和  $\text{arr2}$  的长度为奇数的例子。 $\text{arr1}$  的长度为 5， $\{1, 2, 3, 4, 5\}$  依次表示  $\text{arr1}$  的第 1 个数，第 2 个数……第 5 个数，注意，这个数字表示  $\text{arr1}$  第几个数的意思，并不代表值。 $\text{arr2}$  长度为 5， $\{1', 2', 3', 4', 5'\}$  依次表示  $\text{arr2}$  的第 1 个数，第 2 个数……第 5 个数，注意，这个数字表示  $\text{arr2}$  的第几个数的意思，并不代表值。如果  $\text{arr1}$  的第 3 个数等于  $\text{arr2}$  的第 3 个数 ( $3==3'$ )，那么对这两个数来说，在  $\text{arr1}$  中把 1 和 2 压在底下，在  $\text{arr2}$  中把  $1'$  和  $2'$  压在底下。所以这两个数的值就是上中位数，直接返回  $\text{arr1}[\text{mid1}]$  即可(当然也是  $\text{arr2}[\text{mid2}]$ )。

2)  $\text{arr1}$  和  $\text{arr2}$  的长度为偶数的例子。 $\text{arr1}$  的长度为 4， $\{1, 2, 3, 4\}$  的含义同上。 $\text{arr2}$  的长度为 4， $\{1', 2', 3', 4'\}$  的含义同上。如果  $\text{arr1}$  的第 2 个数等于  $\text{arr2}$  的第 2 个数 ( $2==2'$ )，那么对这两个数来说，在  $\text{arr1}$  中把 1 压在底下，在  $\text{arr2}$  中把  $1'$  压在底下。所以这两个数的值就是上中位数，直接返回  $\text{arr1}[\text{mid1}]$  即可(当然也是  $\text{arr2}[\text{mid2}]$ )。

综上所述，情况一中，如果  $\text{arr1}[\text{mid1}]==\text{arr2}[\text{mid2}]$ ，直接返回  $\text{arr1}[\text{mid1}]$ 。

情况二，如果  $\text{arr1}[\text{mid1}]>\text{arr2}[\text{mid2}]$ 。为了方便理解，仍然举两个例子说明。

1)  $\text{arr1}$  和  $\text{arr2}$  的长度为奇数的例子。 $\text{arr1}$  长度为 5， $\{1, 2, 3, 4, 5\}$  的含义同上。 $\text{arr2}$  长度为 5， $\{1', 2', 3', 4', 5'\}$  的含义同上。如果  $\text{arr1}$  的第 3 个数大于  $\text{arr2}$  的第 3 个数 ( $3>3'$ )，对 4 来说，它可能是第 5 个数吗？不可能。因为在  $\text{arr1}$  中，4 把三个数压在底下，同时又有 ( $3>3'$ )，所以 4 在  $\text{arr2}$  中又起码把三个数压在底下，所以 4 最好情况下是第 7 个数。那么

对 5 来说, 则更不可能。对 2' 来说, 它可能是第 5 个数吗? 不可能。因为在 arr2 中, 2' 只压了一个数, 同时又有  $(3 > 3' \geq 2')$ , 所以 2' 在 arr1 中最多只能把两个数压在底下, 所以 2' 最好情况下是第 4 个数。那么对 1' 来说, 则更不可能。现在我们看一下,  $\{1, 2, 3\}$  和  $\{3', 4', 5'\}$  这两段共同的上中位数, 也就是这 6 个数中第 3 小的数记为 a, 代表什么? a 在  $\{1, 2, 3\}$  和  $\{3', 4', 5'\}$  这两段中, 会把两个数压在下面, 同时也会把原来 arr2 中的 1' 和 2' 压在下面。那么 a 正好就是  $\{1, 2, 3, 4, 5\}$  和  $\{1', 2', 3', 4', 5'\}$  整体第 5 小的数, 也就是想求的结果。所以只要求  $\{1, 2, 3\}$  和  $\{3', 4', 5'\}$  的上中位数即可, 即令  $end1=mid1$ ,  $start2=mid2$ , 然后重复步骤 3。

2) arr1 和 arr2 的长度为偶数的例子。arr1 长度为 4,  $\{1, 2, 3, 4\}$  的含义同上。arr1 长度为 4,  $\{1', 2', 3', 4'\}$  的含义同上。如果 arr1 的第 2 个数大于 arr2 的第 2 个数  $(2 > 2')$ , 对 3 来说, 它可能是第 4 个数吗? 不可能, 因为它起码把四个数压在底下, 最好情况也是第 5 个数, 则 4 更不可能。对 2' 来说, 它可能是第 4 个数吗? 也不可能, 因为它最多只把两个数压在底下, 最好情况也仅是第 3 个数, 则 1' 更不可能。现在我们看一下,  $\{1, 2\}$  和  $\{3', 4'\}$  这两段共同的上中位数, 也就是这 4 个数中第 2 小的数记为 b, 代表什么? b 在  $\{1, 2\}$  和  $\{3', 4'\}$  这两段中, 会把一个数压在下面, 同时也会把原来 arr2 中的 1' 和 2' 压在下面。那么 b 正好就是  $\{1, 2, 3, 4\}$  和  $\{1', 2', 3', 4'\}$  整体第 4 小的数, 也就是想求的结果。所以只要求  $\{1, 2\}$  和  $\{3', 4'\}$  的上中位数即可, 即令  $end1=mid1$ ,  $start2=mid2+1$ , 然后重复步骤 3。

综上所述, 情况二中, 无论怎样, 在 arr1 和 arr2 的范围上都可以二分。

情况三, 如果  $arr1[mid1] < arr2[mid2]$ 。分析方式类似情况二, 这里不再详细解释, 肯定可以二分。arr1 和 arr2 如果长度为奇数, 令  $start1=mid1$ ,  $end2=mid2$ , 然后重复步骤 3。arr1 和 arr2 如果长度为偶数, 令  $start1=mid1+1$ ,  $end2=mid2$ , 然后重复步骤 3。

具体过程请参看如下代码中的 getUpMedian 方法。

```
public int getUpMedian(int[] arr1, int[] arr2) {
    if (arr1 == null || arr2 == null || arr1.length != arr2.length) {
        throw new RuntimeException("Your arr is invalid!");
    }
    int start1 = 0;
    int end1 = arr1.length - 1;
    int start2 = 0;
    int end2 = arr2.length - 1;
    int mid1 = 0;
    int mid2 = 0;
    int offset = 0;
    while (start1 < end1) {
```

```

        mid1 = (start1 + end1) / 2;
        mid2 = (start2 + end2) / 2;
        // 元素个数为奇数，则 offset 为 0，元素个数为偶数，则 offset 为 1。
        offset = ((end1 - start1 + 1) & 1) ^ 1;
        if (arr1[mid1] > arr2[mid2]) {
            end1 = mid1;
            start2 = mid2 + offset;
        } else if (arr1[mid1] < arr2[mid2]) {
            start1 = mid1 + offset;
            end2 = mid2;
        } else {
            return arr1[mid1];
        }
    }
    return Math.min(arr1[start1], arr2[start2]);
}
}

```

## 在两个排序数组中找到第 $K$ 小的数

### 【题目】

给定两个有序数组  $arr1$  和  $arr2$ ，再给定一个整数  $k$ ，返回所有的数中第  $K$  小的数。

### 【举例】

$arr1=[1,2,3,4,5]$ ， $arr2=[3,4,5]$ ， $k=1$ 。

1 是所有数中第 1 小的数，所以返回 1。

$arr1=[1,2,3]$ ， $arr2=[3,4,5,6]$ ， $k=4$ 。

3 是所有数中第 4 小的数，所以返回 3。

### 【要求】

如果  $arr1$  的长度为  $N$ ， $arr2$  的长度为  $M$ ，时间复杂度请达到  $O(\log(\min\{M,N\}))$ ，额外空间复杂度为  $O(1)$ 。

### 【难度】

将 ★★★★★

### 【解答】

在了解本题的解法之前，请读者先阅读上一题“在两个长度相等的排序数组中找到上