

旋转数组的最小数字

题目描述：

把一个数组最开始的若干个元素搬到数组的末尾，我们称之为数组的旋转。 输入一个非递减排序的数组的一个旋转，输出旋转数组的最小元素。 例如数组 {3, 4, 5, 1, 2} 为 {1, 2, 3, 4, 5} 的一个旋转，该数组的最小值为1。 NOTE：给出的所有元素都大于0，若数组大小为0，请返回0。

解决：

(1) 我们用两个指针left, right分别指向数组的第一个元素和最后一个元素。按照题目的旋转的规则，第一个元素应该是大于最后一个元素的（没有重复的元素）。

(2) 找到数组的中间元素。

---中间元素大于第一个元素，则中间元素位于前面的递增子数组，此时最小元素位于中间元素的后 面。我们可以让第一个指针left指向中间元素。移动之后，第一个指针仍然位于前面的递增数组中。

---中间元素小于第一个元素，则中间元素位于后面的递增子数组，此时最小元素位于中间元素的前面。我们可以让第二个指针right指向中间元素。移动之后，第二个指针仍然位于后面的递增数组中。

这样可以缩小寻找的范围。

```
import java.util.ArrayList;
public class Solution {
    public int minNumberInRotateArray(int [] array) {
        if(array.length == 0)
            return 0;
        int left = 0;
        int right = array.length-1;
        int mid =0;
        while(left<right){
            if(right-left == 1){
                mid = right;
                break;
            }
            mid = (left+right)/2;
            //如果左右中指向的元素相等, 那么只能遍历这个局部数组
            if(array[left] == array[right] && array[left] == array[mid]){
                for(int i=left; i<right; i++){
                    if(array[i]>array[i+1])
                        return array[i+1];
                }
            }
            if(array[left] <= array[mid])
                left = mid;
            else
                right =mid;
        }
        return array[mid];
    }
}
```

