

面试题11：旋转数组的最小数字

2018/8/2 20:08:41

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。

C++

```

1 class Solution {
2 public:
3     int minNumberInRotateArray(vector<int>
rotateArray) {
4         if(rotateArray.empty())
5             return 0;
6         int i = rotateArray.size()-1;
7         while(i > 0 && rotateArray[i-1] <=
rotateArray[i])
8             --i;
9         return rotateArray[i];
10    }
11 };

```

Java

```

1 import java.util.ArrayList;
2 public class Solution {
3     public int minNumberInRotateArray(int []
array) {
4         if(array == null || array.length == 0){
5             return 0;
6         }
7         int p1 = 0; //从前往后走
8         int p2 = array.length - 1; //从后往前走
9         int min = array[p1]; //如果没发生旋转，直
接将array[0]的值返回

```

```

10         int mid = 0;
11         //当数组发生旋转
12         while(array[p1] >= array[p2]){
13             //当两个指针走到挨着的位置时，p2就是最小
14             数了
15             if(p2 - p1 == 1){
16                 min = array[p2];
17                 break;
18             }
19             mid = (p1 + p2) / 2;
20             //如果中间位置的数既等于p1位置的数又等于
21             p2位置的数
22             if(array[p1] == array[mid] &&
23             array[p2] == array[mid]){
24                 min = minInorder(array, p1, p2);
25             }
26             if(array[p1] <= array[mid]){ //若中间
27             位置的数位于数组1，让p1走到mid的位置
28             p1 = mid;
29             }else if(array[p2] >= array[mid]){
30             //若中间位置的数位于数组2，让p2走到mid的位置
31             p2 = mid;
32             }
33         }
34         return min;
35     }
36     private int minInorder(int[] array, int p1, int
37     p2){

```

```
32         int min = array[p1];
33         for(int i = p1 + 1; i <= p2; i++){
34             if(min > array[i]){
35                 min = array[i];
36             }
37         }
38         return min;
39     }
40 }
```