

题目

153. 寻找旋转排序数组中的最小值

难度 中等

👍 241



假设按照升序排序的数组在预先未知的某个点上进行了旋转。

(例如，数组 `[0,1,2,4,5,6,7]` 可能变为 `[4,5,6,7,0,1,2]`)。

请找出其中最小的元素。

你可以假设数组中不存在重复元素。

示例 1:

输入: `[3,4,5,1,2]`

输出: `1`

示例 2:

输入: `[4,5,6,7,0,1,2]`

输出: `0`

方法一:

```
1 def findMin(self, nums):
2     """
3     :type nums: List[int]
4     :rtype: int
5     """
6     # 单个元素就直接返回
7     if len(nums) == 1:
8         return nums[0]
9     # 确定左右
10    left = 0
11    right = len(nums) - 1
12
13    # 因为是可能旋转，可能不旋转，不转就是单调增，就直接返回 nums[0]
14    if nums[right] > nums[0]:
```

```

15         return nums[0]
16
17     # 二分查找，其实就是在找一个mid值，mid前是单调增，mid后也是单调值
18     while right >= left:
19         # 中间值
20         mid = (left + right) // 2
21         # 如果mid值比其下一个值大，那么说明 mid + 1 是最小值
22         if nums[mid] > nums[mid + 1]:
23             return nums[mid + 1]
24         # 如果 mid值比其前一个值小，说明mid是最小值
25         if nums[mid - 1] > nums[mid]:
26             return nums[mid]
27         # 所有都不满足，那么0，mid是单调增，那么就在 [mid+1, r] 找最小值
28         if nums[mid] > nums[0]:
29             left = mid + 1
30         否则就在 [l, mid - 1] 找
31     else:
32         right = mid - 1
33

```

方法二：

直接在无序的地方查找

```

1  def findMin(self, nums):
2      """
3      :type nums: List[int]
4      :rtype: int
5      """
6      left = 0
7      right = len(nums) - 1
8
9      while left < right:
10         mid = left + (right - left) // 2
11         # 说明右边无序，所以就直接在右边找
12         if nums[right] < nums[mid]:
13             left = mid + 1
14         else:
15             right = mid
16     return nums[left]

```