3511 398

154. Find Minimum in Rotated Sorted Array II

Suppose an array of length `n` sorted in ascending order is **rotated** between `1` and `n` times. For example, the array `nums = [0,1,4,4,5,6,7]` might become:

- `[4,5,6,7,0,1,4]` if it was rotated `4` times.
- `[0,1,4,4,5,6,7]` if it was rotated `7` times.

Notice that **rotating** an array `[a[0], a[1], a[2], ..., a[n-1]]` 1 time results in the array `[a[n-1], a[0], a[1], a[2], ..., a[n-2]]`.

Given the sorted rotated array `nums` that may contain **duplicates**, return *the minimum element of this array*.

You must decrease the overall operation steps as much as possible.

Example 1:

```
Input: nums = [1,3,5]
Output: 1
```

Example 2:

```
Input: nums = [2,2,2,0,1]
Output: 0
```

Constraints:

- `n == nums.length`
- `1 <= n <= 5000`
- `-5000 <= nums[i] <= 5000`
- `nums` is sorted and rotated between `1` and `n` times.

Follow up: This problem is similar to Find Minimum in Rotated Sorted Array (https://leetcode.com/problems/find-minimum-in-rotated-sorted-array/description/), but `nums` may contain **duplicates**. Would this affect the runtime complexity? How and why?

Related Topics

Similar Questions

Java

```java
class Solution {
    public int findMin(int[] arr) {
                int pivot = findPivot(arr);
          return arr[pivot+1];
              }
    static int findPivot(int[] nums){
        int start = 0;
        int end = nums.length-1;
        while(start<=end){
            int mid = start + (end - start)/2;
```

```
            if(mid<end && nums[mid]>nums[mid+1]){
                return mid;
                }
            if(mid>start && nums[mid]<nums[mid-1]){
                return mid-1;
                }
            if(nums[start]==nums[mid] && nums[mid]==nums[end]){
                if(start<end && nums[start]>nums[start+1]){
                    return start;
                    }
                    start++;
                    if(end>start && nums[end]<nums[end-1]){
                        return end-1;
                        }
                    end--;
                    }
            else if(nums[start] < nums[mid] || (nums[start] == nums[mid] && nums[mid] > nums[end])) {
                start = mid + 1;
                }
            else {
                end = mid - 1;
                 }
            }
            return -1;
        }
}
```

Notes

Custom Testcase (Contribute)

Run Submit

Submission Result: Accepted (/submissions/detail/833475339/) More Details (/submissions/detail/833475339/)

Next challenges:

Find and Replace Pattern (/problems/find-and-replace-pattern) Longest Arithmetic Subsequence (/problems/longest-arithmetic-subsequence) Maximum Length of Subarray With Positive Product (/problems/maximum-length-of-subarray-wi

Share your acceptance!

5

Check out our solution! Reveal Solution (/articles/find-minimum-in-rotated-sorted-array-ii/)