347. Top K Frequent Elements

Medium 8341 353 Add to List Share

Given an integer array `nums` and an integer `k`, return *the* `k` *most frequent elements*. You may return the answer in **any order**.

Example 1:

```
Input: nums = [1,1,1,2,2,3], k = 2
Output: [1,2]
```

Example 2:

```
Input: nums = [1], k = 1
Output: [1]
```

Constraints:

- `1 <= nums.length <= 10^5`
- `k` is in the range `[1, the number of unique elements in the array]`.
- It is **guaranteed** that the answer is **unique**.

Follow up: Your algorithm's time complexity must be better than `O(n log n)`, where n is the array's size.

Accepted 887,706 Submissions 1,368,676

Seen this question in a real interview before? Yes No

Companies

Related Topics

Similar Questions

Java Autocomplete

```java
class Solution {

    public int[] topKFrequent( int[] nums, int k ) {

        HashMap<Integer, Integer> hashMap = new HashMap<>();

        for (int num : nums)
            hashMap.put(num, hashMap.getOrDefault(num, 0) + 1);

        PriorityQueue<Integer> priorityQueue = new PriorityQueue<>(( a, b ) -> hashMap.get(b) - hashMap.get(a));

        priorityQueue.addAll(hashMap.keySet());

        int[] top = new int[k];

        for(int i = 0; i < k; i++){
            top[i] = priorityQueue.poll();
        }

        return top;
    }

}
```

Testcase Run Code Result Debugger

Accepted Runtime: 1 ms

Your input
```
[1,1,1,2,2,3]
2
```

Output `[1,2]` Diff

Expected `[1,2]`