

Description

Solution

Discuss (999+)

Submissions

C++

Autocomplete

347. Top K Frequent Elements

Medium 3689 225 Add to List Share

Given a non-empty array of integers, return the *k* most frequent elements.

Example 1:

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

Example 2:

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

Note:

- You may assume *k* is always valid, $1 \leq k \leq$ number of unique elements.
- Your algorithm's time complexity **must be** better than $O(n \log n)$, where *n* is the array's size.
- It's guaranteed that the answer is unique, in other words the set of the top k frequent elements is unique.
- You can return the answer in any order.

Accepted 461,800 Submissions 750,548

Seen this question in a real interview before? Yes No

Companies

Related Topics

Similar Questions

```
1 class Solution {
2 public:
3     vector<int> topKFrequent(vector<int>& nums, int k) {
4         unordered_map<int,int> hmap;
5         vector<int> ans={};
6         for(int i:nums){
7             ++hmap[i];
8         }
9         vector<vector<int>> buckets(nums.size()+1);
10        for(auto i:hmap){
11            buckets[i.second].push_back(i.first);
12        }
13        for(int i=nums.size();i>=0;i--){
14            vector<int> vec=buckets[i];
15            for(int i:vec){
16                ans.push_back(i);
17                if(ans.size()==k){
18                    break;
19                }
20            }
21            if(ans.size()==k)
22                break;
23        }
24        return ans;
25    }
26};
```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

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

Output [1,2] Diff

Expected [1,2]