

Problem 347: Top K Frequent Elements

Problem Information

Difficulty: Medium

Acceptance Rate: 65.23%

Paid Only: No

Tags: Array, Hash Table, Divide and Conquer, Sorting, Heap (Priority Queue), Bucket Sort, Counting, Quickselect

Problem Description

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]

Example 3:

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

Output: [1,2]

Constraints:

`* `1 <= nums.length <= 105` * `-104 <= nums[i] <= 104` * `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.

Code Snippets

C++:

```
class Solution {  
public:  
vector<int> topKFrequent(vector<int>& nums, int k) {  
  
}  
};
```

Java:

```
class Solution {  
public int[] topKFrequent(int[] nums, int k) {  
  
}  
}
```

Python3:

```
class Solution:  
def topKFrequent(self, nums: List[int], k: int) -> List[int]:
```