# 题目

给你一个整数数组 arr，请你帮忙统计数组中每个数的出现次数。

如果每个数的出现次数都是独一无二的，就返回 true；否则返回 false。

示例 1：

输入：arr = [1,2,2,1,1,3]

输出：true

解释：在该数组中，1 出现了 3 次，2 出现了 2 次，3 只出现了 1 次。没有两个数的出现次数相同。

示例 2：

输入：arr = [1,2]

输出：false

示例 3：

输入：arr = [-3,0,1,-3,1,1,1,-3,10,0]

输出：true

提示：

1 <= arr.length <= 1000

-1000 <= arr[i] <= 1000

# 分析

## 方法一：哈希表

思路：

使用哈希表来统计每个数出现的次数，然后再使用另一个哈希表来记录每个出现次数的频率，最后检查频率是否存在重复即可。

代码：

class Solution {

public:

bool uniqueOccurrences(vector<int>& arr) {

unordered\_map<int, int> freq;

for (int num : arr) {

freq[num]++;

}

unordered\_set<int> seen;

for (const auto& [num, count] : freq) {

if (seen.count(count)) {

return false;

}

seen.insert(count);

}

return true;

}

};