# 题目

给你一个长度为 偶数 的整数数组 nums 。你需要将这个数组分割成 nums1 和 nums2 两部分，要求：

nums1.length == nums2.length == nums.length / 2 。

nums1 应包含 互不相同 的元素。

nums2也应包含 互不相同 的元素。

如果能够分割数组就返回 true ，否则返回 false 。

示例 1：

输入：nums = [1,1,2,2,3,4]

输出：true

解释：分割 nums 的可行方案之一是 nums1 = [1,2,3] 和 nums2 = [1,2,4] 。

示例 2：

输入：nums = [1,1,1,1]

输出：false

解释：分割 nums 的唯一可行方案是 nums1 = [1,1] 和 nums2 = [1,1] 。但 nums1 和 nums2 都不是由互不相同的元素构成。因此，返回 false 。

提示：

1 <= nums.length <= 100

nums.length % 2 == 0

1 <= nums[i] <= 100

# 分析

## 方法一：哈希表

思路：

如果nums[i]的出现次数超过 222，则无法分割，否则可以分割。

代码：

class Solution {

public:

bool isPossibleToSplit(vector<int>& nums) {

unordered\_map<int, int> mp;

for (auto num : nums) {

if (++mp[num] > 2) { // 错误：mp[num]++

return false;

}

}

return true;

}

};

复杂度分析

时间复杂度：O(n)，其中n为nums的长度。

空间复杂度：O(n)。