# 题目

给你一个二维整数数组 nums ，其中 nums[i] 是由 不同 正整数组成的一个非空数组，按 升序排列 返回一个数组，数组中的每个元素在 nums 所有数组 中都出现过。

示例 1：

输入：nums = [[3,1,2,4,5],[1,2,3,4],[3,4,5,6]]

输出：[3,4]

解释：

nums[0] = [3,1,2,4,5]，nums[1] = [1,2,3,4]，nums[2] = [3,4,5,6]，在nums中每个数组中都出现的数字是3和4，所以返回[3,4]。

示例 2：

输入：nums = [[1,2,3],[4,5,6]]

输出：[]

解释：

不存在同时出现在 nums[0] 和 nums[1] 的整数，所以返回一个空列表 [] 。

提示：

1 <= nums.length <= 1000

1 <= sum(nums[i].length) <= 1000

1 <= nums[i][j] <= 1000

nums[i] 中的所有值 互不相同

# 分析

## 方法一：模拟/哈希表

class Solution {

public:

vector<int> intersection(vector<vector<int>>& nums) {

int n = nums.size();

unordered\_set<int> res(nums[0].begin(), nums[0].end());

for (int i = 1; i < n; ++i) {

unordered\_set<int> tmp;

for (int num: nums[i]) {

if (res.count(num)) {

tmp.insert(num);

}

}

res = tmp;

}

vector<int> ans(res.begin(), res.end());

sort(ans.begin(), ans.end());

return ans;

}

};