## 349. 两个数组的交集

▲ 相关企业 Ax

▶ 相关标签

```
给定两个数组 nums1 和 nums2 ,返回 它们的 交集。输出结果中的每个元素一定是 唯一 的。我们可以 不考虑输出结果的顺序。

示例 1:

输入: nums1 = [1,2,2,1], nums2 = [2,2]
输出: [2]

示例 2:

输入: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
输出: [9,4]
输出: [9,4]
解释: [4,9] 也是可通过的
```

```
[] <
⑤ 题目描述 │ ⑤ 解答错误 × │ △ 题解 │ ⑤ 提交记录
                                                         </>/>代码
                                                    0
                                                         C++ ∨ 🔒 智能模式
                                                                                                                            コロコ
← 全部提交记录
                                                            1 class Solution {
解答错误 | 36 / 57 个通过的测试用例
                                          口 官方题解
                                                               public:
                                                                   vector<int> intersection(vector<int>& nums1, vector<int>& nums2)
                                     ☆ 添加到测试用例 🖫
输入
                                                                        sort(nums1.begin(),nums1.end());
                                                            4
 nums1 =
                                                                        sort(nums2.begin(),nums2.end());
 [1,2,2,1]
                                                                       vector<int>intersection;
 nums2 =
                                                            9
                                                                       set_intersection(
 [2,2]
                                                                           nums1.begin(),nums1.end(),
                                                            10
                                                            11
                                                                           nums2.begin(),nums2.end(),
輸出
                                                            12
                                                                           back_inserter(intersection)
 [2,2]
                                                            13
                                                                       );
                                                            14
预期结果
                                                            15
                                                                        return intersection;
                                                           16
 [2]
                                                            17
                                                           18 };
① 不支持可视化
  class Solution {
  public:
     vector<int> intersection(vector<int>& nums1, vector<i
         sort(nums1.begin(),nums1.end());
         sort(nums2.begin(),nums2.end());
        vector<int>intersection;
                       >> 查看更多
                                                         已存储
                                                                                                                               行 18, 列
 添加备注,例如「暴力解法」、「方法—」等
                                                         ☑ 测试用例 │ 〉 测试结果
```

这个题目的错误在于set\_intersection 这个函数不会自动去除重复元素

```
🤾 > 🕒 Test.cpp > ધ Solution > 🔂 intersection(vector<int>&, vector<int>&)
 1 #include <iostream>
 2 #include <vector>
 3 #include <algorithm>
    #include <unordered_set>
5 using namespace std;
    class Solution {
 8 public:
        vector<int> intersection(vector<int>& nums1, vector<int>& nums2) {
             unordered_set<int> set1(nums1.begin(), nums1.end());
10
             unordered_set<int> result;
11
12
             for (int num : nums2) {
13
                 if (set1.count(num)) {
14
15
                     result.insert(num);
17
18
             return vector<int>(result.begin(), result.end());
19
21 };
```

```
int main() {
         Solution solution;
24
25
         vector\langle int \rangle nums1 = \{1, 2, 2, 1\};
26
         vector\langle int \rangle nums2 = {2, 2};
27
28
         vector<int> result = solution.intersection(nums1, nums2);
29
30
         cout << "交集结果: ";
31
         for (int num : result) {
32
              cout << num << " ";
33
34
         cout << endl;</pre>
35
36
37
         return 0;
38
```

利用哈希表解决了这个问题。

- 1. 先记住nums1的数组里面数字的位置。利用哈希表来记住。 并且取名为set1
- 2. 然后在利用范围for来循环nums2的数组里面数字的位置
- 3. 然后检查当前元素num里面的数字是否存在与set1也就是nums1哈希表的相同的数字
- 4. 如果存在,将这个元素插入到resul t中去。

## 为什么使用 unordered\_set?

## 特点:

- 1. 自动去重:
  - 集合中的元素是唯一的,插入相同的值时会自动去重。
- 2. 查找效率高:
  - 对于 unordered\_set ,查找元素的时间复杂度为平均 O(1)。
- 适用于查找操作:
  - 本问题需要判断某个元素是否存在于另一个集合中,用
     unordered\_set 是非常高效的。

```
1 #include<iostream>
 2 #include<unordered_set>
 3 #include<vector>
    using namespace std;
    int main()
 7
        vector<int>nums = {1,2,3,4};
 8
        unordered_set<int>set1(nums.begin(),nums.end());
 9
        unordered_set<int>result;
10
11
        for(int sum : set1)
12
13
            cout<<"合集是:"<<sum<<" ";
14
15
16
        cout<<endl;
        return 0;
17
18
```