

349-两个数组的交集

题述

349. 两个数组的交集

难度 **简单**  474     

给定两个数组，编写一个函数来计算它们的交集。

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]`

说明:

- 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。

思路

通过这道题学习`unordered_set`的用法。这个数据结构类似于我们数学中的集合，最大的特点便是不允许存在重复元素。

题目说明里面进行了强调，输出结果中的每个元素一定是唯一的，也就是说输出的结果是进行了去重的，同时可以不考虑输出结果的顺序。

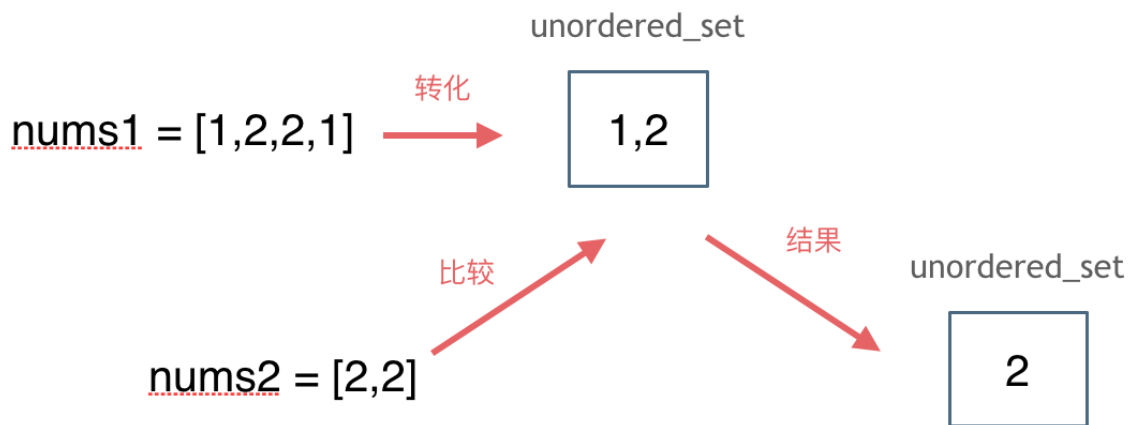
这道题也可以使用数组来做哈希表，可是效率很低，因为使用数组来做哈希的题目，是因为题目都限制了数值的大小。

这道题目没有限制大小，所以无法使用数组，

此外，如果哈希值比较少，特别分散，跨度打，使用数组容易造成空间浪费。

这时需要使用`set`，C++提供了三种`set`

- `set::set`
- `std::multiset`
- `std::unordered_set`



题解

```
class Solution {
public:
    vector<int> intersection(vector<int>& nums1, vector<int>& nums2)
    {
        unordered_set<int> result_set; // 存放结果
        unordered_set<int> nums_set(nums1.begin(), nums1.end());
        for (int num : nums2)
        {
            // 发现nums2的元素 在nums_set里又出现过
            if (nums_set.find(num) != nums_set.end())
            {
                result_set.insert(num);
            }
        }
        return vector<int>(result_set.begin(), result_set.end());
    }
};
```

执行结果: **通过** [显示详情 >](#)

[添加备注](#)

执行用时: **4 ms** , 在所有 C++ 提交中击败了 **89.18%** 的用户

内存消耗: **10.3 MB** , 在所有 C++ 提交中击败了 **36.45%** 的用户

通过测试用例: **55 / 55**

炫耀一下:



[写题解，分享我的解题思路](#)

思考

至于为什么不能直接用set? 而有的题目要用数组:

直接使用set不仅占用空间比数组大，而且速度比数组慢，set把数值映射到key上都要做hash计算的。