

两个数组的交集 II

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: [2,2]

示例 2:

输入: nums1 = [4,9,5], nums2 = [9,4,9,8,4]

输出: [4,9]

```
class Solution {
public:
    vector<int> intersect(vector<int>& nums1, vector<int>& nums2) {
        /*
        //先排序 将两个数组统一为升序序列
        sort(nums1.begin(),nums1.end());
        sort(nums2.begin(),nums2.end());
        vector<int> result;
        int len1=nums1.size();
        int len2=nums2.size();
        //通过索引确定两个指针，分别指向两个序列头部 开始遍历
        int p1=0;
        int p2=0;
        while(p1<len1&& p2<len2)//防止越界 如果其中有一个提前遍历结束，则查询结
束
        {
            //二者比较 由于有序 如果二者不相等 对应指针右移
            if(nums1[p1]<nums2[p2])
            {
                p1++; //默认排序为升序 值小的右移 去找下一个是否相等值
            }
            else if(nums1[p1]>nums2[p2])
            {
                p2++;
            }
            else{
                //找到对应相等的，加入结果数组 两个指针同时右移
                result.push_back(nums1[p1]);
            }
        }
    }
};
```

```
        p1++;
        p2++;
    }
}
return result;
*/
unordered_map<int, int> umap;
vector<int> vec;
for(auto& num : nums1) {
    umap[num]++;
}
for(auto& num : nums2) {
    if(umap.find(num) != umap.end() && umap[num] != 0) { //在 nums1
找到存在于 num2 中的数据，如果此时不为单值则--，直至为单值 单值且已经查找过则退出
        vec.push_back(num);
        umap[num]--;
    }
}
return vec;
}
};
```