# 题目

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。

# 分析

## 方法一：内置函数

**分析：**

放到两个set，用set\_intersection交集

**代码：**

class Solution {

public:

vector<int> intersection(vector<int>& nums1, vector<int>& nums2)

{

vector<int> ans;

set<int> s1(nums1.begin(), nums1.end());

set<int> s2(nums2.begin(), nums2.end());

set\_intersection(s1.begin(),s1.end(), s2.begin(), s2.end(),

insert\_iterator<vector<int>> (ans, ans.begin()));

return ans;

}

};

## 方法二：set+数组

**分析：**

一个放到set，然后遍历另一个数组

**代码：**

class Solution {

public:

vector<int> intersection(vector<int>& nums1, vector<int>& nums2)

{

vector<int> ans;

set<int> s(nums1.begin(), nums1.end());

for(auto &i : nums2){

if(s.erase(i)){

ans.push\_back(i);

}

}

return ans;

}

};

## 方法三：排序+双指针

**分析：**

**代码：**

class Solution {

public:

vector<int> intersection(vector<int>& nums1, vector<int>& nums2)

{

sort(nums1.begin(), nums1.end());

sort(nums2.begin(), nums2.end());

set<int> s;

vector<int>::iterator i = nums1.begin(), j = nums2.begin();

while(i != nums1.end() && j != nums2.end()){

if(\*i == \*j){

s.insert(\*i);

i++;

j++;

}

else if(\*i < \*j){

i++;

}

else{

j++;

}

}

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

return ans;

//如果返回两个集合相同的元素，则直接定义vector<int>返回即可

}

};