

```

1  /*
2  给定两个数组，编写一个函数来计算它们的交集。
3
4  示例 1:
5
6  输入: nums1 = [1,2,2,1], nums2 = [2,2]
7  输出: [2]
8
9  示例 2:
10
11  输入: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
12  输出: [9,4]
13
14  说明:
15
16      输出结果中的每个元素一定是唯一的。
17      我们可以不考虑输出结果的顺序。
18
19  来源：力扣（LeetCode）
20  链接：https://leetcode-cn.com/problems/intersection-of-two-arrays
21  著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
22  */

```

分析:

利用map数据结构分别保存两个数组,然后遍历map1看在map2中是否存在.

方法一:C++\_map

```

1  class solution
2  {
3      public:
4          vector<int> intersection(vector<int>& nums1, vector<int>& nums2)
5          {
6              vector<int>      ret_val      ;
7              map<int,int>      mii1        ;
8              map<int,int>      mii2        ;
9              map<int,int>::iterator  iter;
10             int              i            ;
11             for(i = 0; i < nums1.size();i++)
12             {
13                 mii1[nums1[i]] = 1;
14             }
15             for(i = 0; i < nums2.size();i++)
16             {
17                 mii2[nums2[i]] = 1;
18             }
19
20             iter = mii2.begin();
21             while(iter!=mii2.end())
22             {
23                 if(mii1.count(iter->first)==1)
24                 {

```

```
25         ret_val.push_back(iter->first);
26     }
27     iter++;
28 }
29     return ret_val;
30 }
31 };
32
33 /*
34 执行结果:
35 通过
36 显示详情
37 执行用时 :12 ms, 在所有 C++ 提交中击败了80.21% 的用户
38 内存消耗 :9.8 MB, 在所有 C++ 提交中击败了6.36%的用户
39 */
```

---

AlimyBreak  
2019.10.07