



链滴

## 两个数组的交集 II

作者: [someone31851](#)

原文链接: <https://ld246.com/article/1546684714365>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

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]

### 说明:

- 输出结果中每个元素出现的次数，应与元素在两个数组中出现的次数一致。
- 我们可以不考虑输出结果的顺序。

### 进阶:

- 如果给定的数组已经排好序呢？你将如何优化你的算法？
- 如果 `_nums1_` 的大小比 `_nums2_` 小很多，哪种方法更优？
- 如果 `_nums2_` 的元素存储在磁盘上，磁盘内存是有限的，并且你不能一次加载所有的元素到内存，你该怎么办？

### 思路:

用 Map 集合建立数组元素与次数之间映射，在进行遍历比对

```
class Solution {
    public int[] intersect(int[] nums1, int[] nums2) {
        ArrayList<Integer> temp = new ArrayList<Integer>();
        Map<Integer, Integer> map = new HashMap<Integer, Integer>();
        for (int i = 0; i < nums1.length; i++) {
            Integer val = map.get(nums1[i]);
            if (val != null) {
                val++;
            } else {
                val = 1;
            }
            map.put(nums1[i], val);
        }

        for (int i = 0; i < nums2.length; i++) {
            if (map.containsKey(nums2[i]) && map.get(nums2[i]) != 0) {

                temp.add(nums2[i]);
                map.put(nums2[i], map.get(nums2[i]) - 1);
            }
        }

        int[] result = new int[temp.size()];
```

```
    for (int i = 0; i < temp.size(); i++) {  
        result[j] = temp.get(i);  
    }  
    return result;  
}  
}
```