# 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: [2,2]

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中，你该怎么办？

# 分析

## 方法一：哈希映射

前面的问题 349. 两个数组的交集，我们使用 set 来实现线性时间复杂度。在这里，我们需要使用 HashMap 来跟踪每个数字出现的次数。

我们先在 HashMap 记录一个数组中的存在的数字和对应出现的次数。然后，我们遍历第二个数组，检查数字在 HashMap 中是否存在，如果存在且计数为正，则将该数字添加到答案并减少 HashMap 中的计数。

检查数组的大小并对较小的数组进行哈希映射是一个小细节，当其中一个数组较大时，会减少内存的使用。

如果 nums1 元素个数大于 nums2，则交换数组元素。

对于 nums1 的每个元素，添加到 HashMap m 中，如果元素已经存在则增加对应的计数。

初始化 k = 0，记录当前交集元素个数。

遍历数组 nums2：

检查元素在 m 是否存在，若存在且计数为正：

将元素拷贝到 nums1[k]，且 k++。

减少 m 中对应元素的计数。

返回 nums1 前 k 个元素。

## 方法二：排序

当输入数据是有序的，推荐使用此方法。在这里，我们对两个数组进行排序，并且使用两个指针在一次扫面找出公共的数字。

算法：

对数组 nums1 和 nums2 排序。

初始化指针 i，j 和 k 为 0。

指针 i 指向 nums1，指针 j 指向 nums2：

如果 nums1[i] < nums2[j]，则 i++。

如果 nums1[i] > nums2[j]，则 j++。

如果 nums1[i] == nums2[j]，将元素拷贝到 nums1[k]，且 i++，j++，k++。

返回数组 nums1 前 k 个元素。

## 方法三：内置函数