Problem 350: Intersection of Two Arrays II

Problem Information

Difficulty: Easy
Acceptance Rate: 59.40%
Paid Only: No
Tags: Array, Hash Table, Two Pointers, Binary Search, Sorting

Problem Description

Given two integer arrays `nums1` and `nums2`, return _an array of their intersection_. Each element in the result must appear as many times as it shows in both arrays and you may return the result in **any order**.

Example 1:

Input: nums1 = [1,2,2,1], nums2 = [2,2] **Output:** [2,2]

Example 2:

Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4] **Output:** [4,9] **Explanation:** [9,4] is also accepted.

Constraints:

* `1 <= nums1.length, nums2.length <= 1000` * `0 <= nums1[i], nums2[i] <= 1000`

Follow up:

* What if the given array is already sorted? How would you optimize your algorithm? * What if `nums1`'s size is small compared to `nums2`'s size? Which algorithm is better? * What if elements of `nums2` are stored on disk, and the memory is limited such that you cannot load all elements into the memory at once?

Code Snippets

C++:

```
class Solution {
public:
vector<int> intersect(vector<int>& nums1, vector<int>& nums2) {

}
};
```

Java:

```
class Solution {
public int[] intersect(int[] nums1, int[] nums2) {

}
}
```

Python3:

```
class Solution:
def intersect(self, nums1: List[int], nums2: List[int]) -> List[int]:
```