Problem 2248: Intersection of Multiple Arrays

Problem Information

Difficulty: Easy
Acceptance Rate: 68.58%
Paid Only: No
Tags: Array, Hash Table, Sorting, Counting

Problem Description

Given a 2D integer array `nums` where `nums[i]` is a non-empty array of **distinct** positive integers, return _the list of integers that are present in**each array** of_ `nums` _sorted in**ascending order**_.

Example 1:

Input: nums = [[_**3**_ ,1,2,_**4**_ ,5],[1,2,_**3**_ ,_**4**_],[_**3**_ ,_**4**_ ,5,6]] **Output:** [3,4] **Explanation:** The only integers present in each of nums[0] = [_**3**_ ,1,2,_**4**_ ,5], nums[1] = [1,2,_**3**_ ,_**4**_], and nums[2] = [_**3**_ ,_**4**_ ,5,6] are 3 and 4, so we return [3,4].

Example 2:

Input: nums = [[1,2,3],[4,5,6]] **Output:** [] **Explanation:** There does not exist any integer present both in nums[0] and nums[1], so we return an empty list [].

Constraints:

* `1 <= nums.length <= 1000` * `1 <= sum(nums[i].length) <= 1000` * `1 <= nums[i][j] <= 1000` * All the values of `nums[i]` are **unique**.

Code Snippets

C++:

```
class Solution {
public:
vector<int> intersection(vector<vector<int>>& nums) {

}
};
```

Java:

```
class Solution {
public List<Integer> intersection(int[][] nums) {

}
}
```

Python3:

```
class Solution:
def intersection(self, nums: List[List[int]]) -> List[int]:
```