Problem 2032: Two Out of Three

Problem Information

Difficulty: Easy
Acceptance Rate: 77.12%
Paid Only: No
Tags: Array, Hash Table, Bit Manipulation

Problem Description

Given three integer arrays `nums1`, `nums2`, and `nums3`, return _a**distinct** array containing all the values that are present in **at least two** out of the three arrays. You may return the values in **any** order_.

Example 1:

Input: nums1 = [1,1,3,2], nums2 = [2,3], nums3 = [3] **Output:** [3,2] **Explanation:** The values that are present in at least two arrays are: - 3, in all three arrays. - 2, in nums1 and nums2.

Example 2:

Input: nums1 = [3,1], nums2 = [2,3], nums3 = [1,2] **Output:** [2,3,1] **Explanation:** The values that are present in at least two arrays are: - 2, in nums2 and nums3. - 3, in nums1 and nums2. - 1, in nums1 and nums3.

Example 3:

Input: nums1 = [1,2,2], nums2 = [4,3,3], nums3 = [5] **Output:** [] **Explanation:** No value is present in at least two arrays.

Constraints:

* `1 <= nums1.length, nums2.length, nums3.length <= 100` * `1 <= nums1[i], nums2[j], nums3[k] <= 100`

Code Snippets

C++:

```
class Solution {
public:
vector<int> twoOutOfThree(vector<int>& nums1, vector<int>& nums2,
vector<int>& nums3) {

}
};
```

Java:

```
class Solution {
public List<Integer> twoOutOfThree(int[] nums1, int[] nums2, int[] nums3) {

}
}
```

Python3:

```
class Solution:
def twoOutOfThree(self, nums1: List[int], nums2: List[int], nums3: List[int])
-> List[int]:
```