6041. Intersection of Multiple Arrays

My Submissions (/contest/weekly-contest-290/problems/intersection-of-multiple-arrays/submissions/) Back to Contest (/contest/weekly-contest-290/)

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***.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Easy

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] = [
```

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**.

JavaScript

```
const intersection = (g) => {
    let se = new Set();
    g = g.map(a => {
        let m = new Map();
        for (let i = 0; i < a.length; i++) {
            m.set(a[i], i);
            se.add(a[i]);
        }
        return m;
    });
    let u = [...se], res = [];
    u.sort((x, y) => x - y);
    for (const x of u) {
        if (inAll(x, g)) res.push(x);
    }
    return res;
};

const inAll = (x, g) => {
    for (const m of g) {
        if (!m.has(x)) return false;
    }
    return true;
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/686322992/) More Details > (/submissions/detail/686322992/)

Share your acceptance!