Back to Contest (/contest/weekly-contest-333/)

You are given two 2D integer arrays nums1 and nums2.

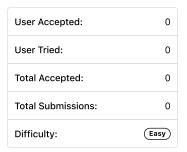
- $nums1[i] = [id_i, val_i]$ indicate that the number with the id_i has a value equal to val_i .
- $nums2[i] = [id_i, val_i]$ indicate that the number with the id id_i has a value equal to val_i .

Each array contains unique ids and is sorted in ascending order by id.

Merge the two arrays into one array that is sorted in ascending order by id, respecting the following conditions:

- · Only ids that appear in at least one of the two arrays should be included in the resulting array.
- Each id should be included only once and its value should be the sum of the values of this id in the two arrays. If the id does not exist in one of the two arrays then its value in that array is considered to be $\, 0 \,$.

Return the resulting array. The returned array must be sorted in ascending order by id.



Example 1:

```
Input: nums1 = [[1,2],[2,3],[4,5]], nums2 = [[1,4],[3,2],[4,1]]
Output: [[1,6],[2,3],[3,2],[4,6]]
Explanation: The resulting array contains the following:
- id = 1, the value of this id is 2 + 4 = 6.
- id = 2, the value of this id is 3.
- id = 3, the value of this id is 2.
- id = 4, the value of this id is 5 + 1 = 6.
```

Example 2:

```
Input: nums1 = [[2,4],[3,6],[5,5]], nums2 = [[1,3],[4,3]]
Output: [[1,3],[2,4],[3,6],[4,3],[5,5]]
Explanation: There are no common ids, so we just include each id with its value in the resulting list.
```

Constraints:

- 1 <= nums1.length, nums2.length <= 200
- nums1[i].length == nums2[j].length == 2
- 1 <= id_i, val_i <= 1000
- · Both arrays contain unique ids.
- · Both arrays are in strictly ascending order by id.

```
JavaScript
                                                                                                                                 क
                                                                                                                                        \mathfrak{C}
1
    const add0ne0rManyMap = (m, x, cnt = 1) \Rightarrow m.set(x, m.get(x) + cnt || cnt);
 2
3
    const mergeArrays = (a, b) \Rightarrow \{
 4
         let m = new Map();
         for (const [id, v] of a) addOneOrManyMap(m, id, v);
 5
 6
         for (const [id, v] of b) addOneOrManyMap(m, id, v);
7
         let res = Array.from(m.entries()).sort((x, y) \Rightarrow x[0] - y[0]);
 8
         return res;
9
    };
10
```