6363. Convert an Array Into a 2D Array With Conditions

My Submissions (/contest/weekly-contest-339/problems/convert-an-array-into-a-2d-array-with-conditions/submissions/)

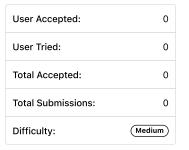
Back to Contest (/contest/weekly-contest-339/)

You are given an integer array nums . You need to create a 2D array from nums satisfying the following conditions:

- The 2D array should contain **only** the elements of the array nums .
- Each row in the 2D array contains distinct integers.
- The number of rows in the 2D array should be minimal.

Return the resulting array. If there are multiple answers, return any of them.

Note that the 2D array can have a different number of elements on each row.



€

Example 1:

```
Input: nums = [1,3,4,1,2,3,1]
Output: [[1,3,4,2],[1,3],[1]]
Explanation: We can create a 2D array that contains the following rows:
- 1,3,4,2
- 1,3
- 1
All elements of nums were used, and each row of the 2D array contains distinct integers, so it is a valid answer.
It can be shown that we cannot have less than 3 rows in a valid array.
```

Example 2:

```
Input: nums = [1,2,3,4]
Output: [[4,3,2,1]]
Explanation: All elements of the array are distinct, so we can keep all of them in the first row of the 2D array.
```

Constraints:

- 1 <= nums.length <= 200
- 1 <= nums[i] <= nums.length

```
JavaScript
                                                                                                                       ₼ 2 
1 v const findMatrix = (a) ⇒ {
        let N = Math.max(...a), f = Array(N + 1).fill(0), res = [], d = [];
3
        for (const x of a) f[x]++;
 4
        for (let i = 0; i \le N; i++) d.push([i, f[i]]);
5 •
        while (1) {
 6
             d.sort((x, y) \Rightarrow y[1] - x[1]);
 7
             if (d[0][1] == 0) break;
8
            let cur = [];
9 •
             for (let i = 0; i <= N; i++) {
10 🕶
                 if (d[i][1] > 0) {
11
                     cur.push(d[i][0]);
12
                     d[i][1]--;
13 ▼
                 } else {
14
                     break;
15
16
17
             res.push(cur);
18
19
        return res;
20
    };
```

