

5554. Check Array Formation Through Concatenation

[My Submissions \(/contest/weekly-contest-213/problems/check-array-formation-through-concatenation/submissions/\)](#)
[Back to Contest \(/contest/weekly-contest-213/\)](#)

You are given an array of **distinct** integers `arr` and an array of integer arrays `pieces`, where the integers in `pieces` are **distinct**. Your goal is to form `arr` by concatenating the arrays in `pieces` **in any order**. However, you are **not** allowed to reorder the integers in each array `pieces[i]`.

Return `true` if it is possible to form the array `arr` from `pieces`. Otherwise, return `false`.

Example 1:

Input: `arr = [85], pieces = [[85]]`
Output: `true`

Example 2:

Input: `arr = [15,88], pieces = [[88],[15]]`
Output: `true`
Explanation: Concatenate `[15]` then `[88]`

Example 3:

Input: `arr = [49,18,16], pieces = [[16,18,49]]`
Output: `false`
Explanation: Even though the numbers match, we cannot reorder `pieces[0]`.

Example 4:

Input: `arr = [91,4,64,78], pieces = [[78],[4,64],[91]]`
Output: `true`
Explanation: Concatenate `[91]` then `[4,64]` then `[78]`

Example 5:

Input: `arr = [1,3,5,7], pieces = [[2,4,6,8]]`
Output: `false`

Constraints:

- `1 <= pieces.length <= arr.length <= 100`
- `sum(pieces[i].length) == arr.length`
- `1 <= pieces[i].length <= arr.length`
- `1 <= arr[i], pieces[i][j] <= 100`
- The integers in `arr` are **distinct**.
- The integers in `pieces` are **distinct** (i.e., If we flatten `pieces` in a 1D array, all the integers in this array are distinct).

JavaScript



```

1 /**
2  * @param {number[]} arr
3  * @param {number[][]} pieces
4  * @return {boolean}
5  */
6 const canFormArray = (arr, pieces) => {
7   let m = new Map();
8   for (let i = 0; i < arr.length; i++) {
9     m.set(arr[i], i);
10  }

```

```
11▼ for (let i = 0; i < pieces.length; i++) {
12▼   if (pieces[i].length == 1 && m.has(pieces[i][0])) {
13     continue;
14▼   } else if (pieces[i].length > 1 && m.has(pieces[i][0])) {
15     let idx = m.get(pieces[i][0]);
16     idx++;
17     if (idx >= arr.length) return false;
18▼     for (let j = 1; j < pieces[i].length; j++) {
19▼       if (arr[idx] == pieces[i][j]) {
20         idx++;
21         if (idx >= arr.length && j < pieces[i].length - 1) return false;
22▼       } else {
23         return false;
24       }
25     }
26▼   } else {
27     return false;
28   }
29 }
30 return true;
31 };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)Submission Result: **Accepted** (/submissions/detail/415423686/) ?[More Details > \(/submissions/detail/415423686/\)](#)

Share your acceptance!