Problem 1640: Check Array Formation Through Concatenation

Problem Information

Difficulty: Easy
Acceptance Rate: 57.15%
Paid Only: No
Tags: Array, Hash Table

Problem Description

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 = [15,88], pieces = [[88],[15]] **Output:** true **Explanation:** Concatenate [15] then [88]

Example 2:

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

Example 3:

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

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).

Code Snippets

C++:

```
class Solution {
public:
bool canFormArray(vector<int>& arr, vector<vector<int>>& pieces) {

}
};
```

Java:

```
class Solution {
public boolean canFormArray(int[] arr, int[][] pieces) {

}
}
```

Python3:

```
class Solution:
def canFormArray(self, arr: List[int], pieces: List[List[int]]) -> bool:
```