

能否连接形成数组

给你一个整数数组 `arr`，数组中的每个整数 **互不相同**。另有一个由整数数组构成的数组 `pieces`，其中的整数也 **互不相同**。请你以 **任意顺序** 连接 `pieces` 中的数组以形成 `arr`。但是，**不允许** 对每个数组 `pieces[i]` 中的整数重新排序。

如果可以连接 `pieces` 中的数组形成 `arr`，返回 `true`；否则，返回 `false`。

示例 1:

输入: `arr = [85], pieces = [[85]]`

输出: `true`

示例 2:

输入: `arr = [15,88], pieces = [[88],[15]]`

输出: `true`

解释: 依次连接 `[15]` 和 `[88]`

示例 3:

输入: `arr = [49,18,16], pieces = [[16,18,49]]`

输出: `false`

解释: 即便数字相符，也不能重新排列 `pieces[0]`

示例 4:

输入: `arr = [91,4,64,78], pieces = [[78],[4,64],[91]]`

输出: `true`

解释: 依次连接 `[91]`、`[4,64]` 和 `[78]`

示例 5:

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

输出: `false`

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

```
    for i in arr:
        for j in pieces:
            if i==j[0]:#只要arr中元素能在piece中子列表中第一个元素对应则就
添加 不破坏子列表顺序
                result+=j
    return arr==result
```