Problem 1460: Make Two Arrays Equal by Reversing Subarrays

Problem Information

Difficulty: Easy
Acceptance Rate: 75.81%
Paid Only: No
Tags: Array, Hash Table, Sorting

Problem Description

You are given two integer arrays of equal length `target` and `arr`. In one step, you can select any **non-empty subarray** of `arr` and reverse it. You are allowed to make any number of steps.

Return `true` _if you can make_`arr` _equal to_`target` _ or _`false` _otherwise_.

Example 1:

Input: target = [1,2,3,4], arr = [2,4,1,3] **Output:** true **Explanation:** You can follow the next steps to convert arr to target: 1- Reverse subarray [2,4,1], arr becomes [1,4,2,3] 2- Reverse subarray [4,2], arr becomes [1,2,4,3] 3- Reverse subarray [4,3], arr becomes [1,2,3,4] There are multiple ways to convert arr to target, this is not the only way to do so.

Example 2:

Input: target = [7], arr = [7] **Output:** true **Explanation:** arr is equal to target without any reverses.

Example 3:

Input: target = [3,7,9], arr = [3,7,11] **Output:** false **Explanation:** arr does not have value 9 and it can never be converted to target.

Constraints:

* `target.length == arr.length` * `1 <= target.length <= 1000` * `1 <= target[i] <= 1000` * `1 <= arr[i] <= 1000`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public boolean canBeEqual(int[] target, int[] arr) {

}
}
```

Python3:

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