Problem 822: Card Flipping Game

Problem Information

Difficulty: Medium
Acceptance Rate: 49.32%
Paid Only: No
Tags: Array, Hash Table

Problem Description

You are given two **0-indexed** integer arrays `fronts` and `backs` of length `n`, where the `ith` card has the positive integer `fronts[i]` printed on the front and `backs[i]` printed on the back. Initially, each card is placed on a table such that the front number is facing up and the other is facing down. You may flip over any number of cards (possibly zero).

After flipping the cards, an integer is considered **good** if it is facing down on some card and **not** facing up on any card.

Return _the minimum possible good integer after flipping the cards_. If there are no good integers, return `0`.

Example 1:

Input: fronts = [1,2,4,4,7], backs = [1,3,4,1,3] **Output:** 2 **Explanation:** If we flip the second card, the face up numbers are [1,3,4,4,7] and the face down are [1,2,4,1,3]. 2 is the minimum good integer as it appears facing down but not facing up. It can be shown that 2 is the minimum possible good integer obtainable after flipping some cards.

Example 2:

Input: fronts = [1], backs = [1] **Output:** 0 **Explanation:** There are no good integers no matter how we flip the cards, so we return 0.

Constraints:

* `n == fronts.length == backs.length` * `1 <= n <= 1000` * `1 <= fronts[i], backs[i] <= 2000`

Code Snippets

C++:

```
class Solution {
public:
int flipgame(vector<int>& fronts, vector<int>& backs) {

}
};
```

Java:

```
class Solution {
public int flipgame(int[] fronts, int[] backs) {

}
}
```

Python3:

```
class Solution:
def flipgame(self, fronts: List[int], backs: List[int]) -> int:
```