3785. Minimum Swaps to Avoid Forbidden Values

Solved

Hard Topics Hint

You are given two integer arrays, `nums` and `forbidden`, each of length `n`.

You may perform the following operation any number of times (including zero):

- Choose two **distinct** indices `i` and `j`, and swap `nums[i]` with `nums[j]`.

Return the **minimum** number of swaps required such that, for every index `i`, the value of `nums[i]` is **not equal** to `forbidden[i]`. If no amount of swaps can ensure that every index avoids its forbidden value, return -1.

Example 1:

> **Input:** nums = [1,2,3], forbidden = [3,2,1]
>
> **Output:** 1
>
> **Explanation:**
>
> One optimal set of swaps:
>
> - Select indices `i = 0` and `j = 1` in `nums` and swap them, resulting in `nums = [2, 1, 3]`.
> - After this swap, for every index `i`, `nums[i]` is not equal to `forbidden[i]`.

Example 2:

> **Input:** nums = [4,6,6,5], forbidden = [4,6,5,5]
>
> **Output:** 2
>
> **Explanation:**
>
> One optimal set of swaps:
> - Select indices `i = 0` and `j = 2` in `nums` and swap them, resulting in `nums = [6, 6, 4, 5]`.
> - Select indices `i = 1` and `j = 3` in `nums` and swap them, resulting in `nums = [6, 5, 4, 6]`.
> - After these swaps, for every index `i`, `nums[i]` is not equal to `forbidden[i]`.

Example 3:

> **Input:** nums = [7,7], forbidden = [8,7]
>
> **Output:** -1
>
> **Explanation:**
>
> It is not possible to make `nums[i]` different from `forbidden[i]` for all indices.

Example 4:

> **Input:** nums = [1,2], forbidden = [2,1]
>
> **Output:** 0
>
> **Explanation:**
>
> No swaps are required because `nums[i]` is already different from `forbidden[i]` for all indices, so the answer is 0.

Constraints:

- `1 <= n == nums.length == forbidden.length <= 10^5`
- `1 <= nums[i], forbidden[i] <= 10^9`

Seen this question in a real interview before? 1/5

Yes No

Accepted 8,888 /31.1K | Acceptance Rate 28.6%

Topics

Hint 1

Hint 2

Hint 3

Hint 4

Hint 5

Hint 6

Discussion (33)