You are given a **0-indexed** array nums that consists of n **distinct** positive integers. Apply m operations to this array, where in the ith operation you replace the number operations[i][0] with operations[i][1].

It is guaranteed that in the ith operation:

* operations[i][0] **exists** in nums.
* operations[i][1] does **not** exist in nums.

Return *the array obtained after applying all the operations*.

**Example 1:**

**Input:** nums = [1,2,4,6], operations = [[1,3],[4,7],[6,1]]

**Output:** [3,2,7,1]

**Explanation:** We perform the following operations on nums:

- Replace the number 1 with 3. nums becomes [**3**,2,4,6].

- Replace the number 4 with 7. nums becomes [3,2,**7**,6].

- Replace the number 6 with 1. nums becomes [3,2,7,**1**].

We return the final array [3,2,7,1].

**Example 2:**

**Input:** nums = [1,2], operations = [[1,3],[2,1],[3,2]]

**Output:** [2,1]

**Explanation:** We perform the following operations to nums:

- Replace the number 1 with 3. nums becomes [**3**,2].

- Replace the number 2 with 1. nums becomes [3,**1**].

- Replace the number 3 with 2. nums becomes [**2**,1].

We return the array [2,1].

**Constraints:**

* n == nums.length
* m == operations.length
* 1 <= n, m <= 105
* All the values of nums are **distinct**.
* operations[i].length == 2
* 1 <= nums[i], operations[i][0], operations[i][1] <= 106
* operations[i][0] will exist in nums when applying the ith operation.
* operations[i][1] will not exist in nums when applying the ith operation.