3080. Mark Elements on Array by Performing Queries

Description

You are given a **0-indexed** array `nums` of size `n` consisting of positive integers.

You are also given a 2D array `queries` of size `m` where `queries[i] = [index_i, k_i]`.

Initially all elements of the array are **unmarked**.

You need to apply `m` queries on the array in order, where on the `i^th` query you do the following:

- Mark the element at index `index_i` if it is not already marked.
- Then mark `k_i` unmarked elements in the array with the **smallest** values. If multiple such elements exist, mark the ones with the smallest indices. And if less than `k_i` unmarked elements exist, then mark all of them.

Return *an array answer of size* `m` *where* `answer[i]` *is the* ***sum*** *of unmarked elements in the array after the* `i^th` *query*.

Example 1:

Input: `nums = [1,2,2,1,2,3,1], queries = [[1,2],[3,3],[4,2]]`

Output: `[8,3,0]`

Explanation:

We do the following queries on the array:

- Mark the element at index `1`, and `2` of the smallest unmarked elements with the smallest indices if they exist, the marked elements now are `nums = [1,2,2,1,2,3,1]`. The sum of unmarked elements is `2 + 2 + 3 + 1 = 8`.
- Mark the element at index `3`, since it is already marked we skip it. Then we mark `3` of the smallest unmarked elements with the smallest indices, the marked elements now are `nums = [1,2,2,1,2,3,1]`. The sum of unmarked elements is `3`.
- Mark the element at index `4`, since it is already marked we skip it. Then we mark `2` of the smallest unmarked elements with the smallest indices if they exist, the marked elements now are `nums = [1,2,2,1,2,3,1]`. The sum of unmarked elements is `0`.

Example 2:

Input: `nums = [1,4,2,3], queries = [[0,1]]`

Output: `[7]`

Explanation: We do one query which is mark the element at index `0` and mark the smallest element among unmarked elements. The marked elements will be `nums = [1,4,2,3]`, and the sum of unmarked elements is `4 + 3 = 7`.

Constraints:

- `n == nums.length`
- `m == queries.length`
- `1 <= m <= n <= 10^5`
- `1 <= nums[i] <= 10^5`
- `queries[i].length == 2`
- `0 <= index_i, k_i <= n - 1`