# Problem 3080: Mark Elements on Array by Performing Queries

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 48.42%
**Paid Only:** No
**Tags:** Array, Hash Table, Sorting, Heap (Priority Queue), Simulation

## Problem 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] = [indexi, ki]`.

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

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

* Mark the element at index `indexi` if it is not already marked. * Then mark `ki` 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 `ki` 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 _`ith` _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 <= 105` * `1 <= nums[i] <= 105` * `queries[i].length == 2` * `0 <= indexi, ki <= n - 1`

## Code Snippets

**C++:**

```
class Solution {
public:
vector<long long> unmarkedSumArray(vector<int>& nums, vector<vector<int>>&
queries) {

}
};
```

**Java:**

```
class Solution {
public long[] unmarkedSumArray(int[] nums, int[][] queries) {

}
}
```

**Python3:**

```
class Solution:
def unmarkedSumArray(self, nums: List[int], queries: List[List[int]]) ->
List[int]:
```