

2343. Query Kth Smallest Trimmed Number

Description

You are given a **0-indexed** array of strings `nums`, where each string is of **equal length** and consists of only digits.

You are also given a **0-indexed** 2D integer array `queries` where `queries[i] = [ki, trimi]`. For each `queries[i]`, you need to:

- **Trim** each number in `nums` to its **rightmost** `trimi` digits.
- Determine the **index** of the `kith` smallest trimmed number in `nums`. If two trimmed numbers are equal, the number with the **lower** index is considered to be smaller.
- Reset each number in `nums` to its original length.

Return *an array* `answer` *of the same length as* `queries`, *where* `answer[i]` *is the answer to the* `ith` *query.*

Note :

- To trim to the rightmost `x` digits means to keep removing the leftmost digit, until only `x` digits remain.
- Strings in `nums` may contain leading zeros.

Example 1:

Input: `nums = ["102","473","251","814"], queries = [[1,1],[2,3],[4,2],[1,2]]`
Output: `[2,2,1,0]`
Explanation:
1. After trimming to the last digit, `nums = ["2","3","1","4"]`. The smallest number is 1 at index 2.
2. Trimmed to the last 3 digits, `nums` is unchanged. The 2nd smallest number is 251 at index 2.
3. Trimmed to the last 2 digits, `nums = ["02","73","51","14"]`. The 4th smallest number is 73.
4. Trimmed to the last 2 digits, the smallest number is 2 at index 0.
Note that the trimmed number "02" is evaluated as 2.

Example 2:

Input: `nums = ["24","37","96","04"], queries = [[2,1],[2,2]]`
Output: `[3,0]`
Explanation:
1. Trimmed to the last digit, `nums = ["4","7","6","4"]`. The 2nd smallest number is 4 at index 3.
There are two occurrences of 4, but the one at index 0 is considered smaller than the one at index 3.
2. Trimmed to the last 2 digits, `nums` is unchanged. The 2nd smallest number is 24.

Constraints:

- `1 <= nums.length <= 100`
- `1 <= nums[i].length <= 100`
- `nums[i]` consists of only digits.
- All `nums[i].length` are **equal**.
- `1 <= queries.length <= 100`
- `queries[i].length == 2`
- `1 <= ki <= nums.length`
- `1 <= trimi <= nums[i].length`

Follow up: Could you use the **Radix Sort Algorithm** to solve this problem? What will be the complexity of that solution?

