Problem 2653: Sliding Subarray Beauty

Problem Information

Difficulty: Medium
Acceptance Rate: 35.38%
Paid Only: No
Tags: Array, Hash Table, Sliding Window

Problem Description

Given an integer array `nums` containing `n` integers, find the **beauty** of each subarray of size `k`.

The **beauty** of a subarray is the `xth`**smallest integer** in the subarray if it is **negative** , or `0` if there are fewer than `x` negative integers.

Return _an integer array containing_`n - k + 1` _integers, which denote the_**beauty** _of the subarrays**in order** from the first index in the array._

* A subarray is a contiguous **non-empty** sequence of elements within an array.

Example 1:

Input: nums = [1,-1,-3,-2,3], k = 3, x = 2 **Output:** [-1,-2,-2] **Explanation:** There are 3 subarrays with size k = 3. The first subarray is [1, -1, -3] and the 2nd smallest negative integer is -1. The second subarray is [-1, -3, -2] and the 2nd smallest negative integer is -2. The third subarray is [-3, -2, 3] and the 2nd smallest negative integer is -2.

Example 2:

Input: nums = [-1,-2,-3,-4,-5], k = 2, x = 2 **Output:** [-1,-2,-3,-4] **Explanation:** There are 4 subarrays with size k = 2. For [-1, -2], the 2nd smallest negative integer is -1. For [-2, -3], the 2nd smallest negative integer is -2. For [-3, -4], the 2nd smallest negative integer is -3. For [-4, -5], the 2nd smallest negative integer is -4.

Example 3:

Input: nums = [-3,1,2,-3,0,-3], k = 2, x = 1 **Output:** [-3,0,-3,-3,-3] **Explanation:** There are 5 subarrays with size k = 2**.** For [-3, 1], the 1st smallest negative integer is -3. For [1, 2], there is no negative integer so the beauty is 0. For [2, -3], the 1st smallest negative integer is -3. For [-3, 0], the 1st smallest negative integer is -3. For [0, -3], the 1st smallest negative integer is -3.

Constraints:

* `n == nums.length ` * `1 <= n <= 105` * `1 <= k <= n` * `1 <= x <= k ` * `-50 <= nums[i] <= 50 `

Code Snippets

C++:

```
class Solution {
public:
vector<int> getSubarrayBeauty(vector<int>& nums, int k, int x) {

}
};
```

Java:

```
class Solution {
public int[] getSubarrayBeauty(int[] nums, int k, int x) {

}
}
```

Python3:

```
class Solution:
def getSubarrayBeauty(self, nums: List[int], k: int, x: int) -> List[int]:
```