Problem 2519: Count the Number of K-Big Indices

Problem Information

Difficulty: Hard
Acceptance Rate: 53.44%
Paid Only: Yes
Tags: Array, Binary Search, Divide and Conquer, Binary Indexed Tree, Segment Tree, Merge Sort, Ordered Set

Problem Description

You are given a **0-indexed** integer array `nums` and a positive integer `k`.

We call an index `i` **k-big** if the following conditions are satisfied:

* There exist at least `k` different indices `idx1` such that `idx1 < i` and `nums[idx1] < nums[i]`.
* There exist at least `k` different indices `idx2` such that `idx2 > i` and `nums[idx2] < nums[i]`.

Return _the number of k-big indices_.

Example 1:

Input: nums = [2,3,6,5,2,3], k = 2 **Output:** 2 **Explanation:** There are only two 2-big indices in nums: - i = 2 --> There are two valid idx1: 0 and 1. There are three valid idx2: 2, 3, and 4. - i = 3 --> There are two valid idx1: 0 and 1. There are two valid idx2: 3 and 4.

Example 2:

Input: nums = [1,1,1], k = 3 **Output:** 0 **Explanation:** There are no 3-big indices in nums.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i], k <= nums.length`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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