Problem 3255: Find the Power of K-Size Subarrays II

Problem Information

Difficulty: Medium
Acceptance Rate: 31.02%
Paid Only: No
Tags: Array, Sliding Window

Problem Description

You are given an array of integers `nums` of length `n` and a _positive_ integer `k`.

The **power** of an array is defined as:

* Its **maximum** element if _all_ of its elements are **consecutive** and **sorted** in **ascending** order. * -1 otherwise.

You need to find the **power** of all subarrays of `nums` of size `k`.

Return an integer array `results` of size `n - k + 1`, where `results[i]` is the _power_ of `nums[i..(i + k - 1)]`.

Example 1:

Input: nums = [1,2,3,4,3,2,5], k = 3

Output: [3,4,-1,-1,-1]

Explanation:

There are 5 subarrays of `nums` of size 3:

* `[1, 2, 3]` with the maximum element 3. * `[2, 3, 4]` with the maximum element 4. * `[3, 4, 3]` whose elements are **not** consecutive. * `[4, 3, 2]` whose elements are **not** sorted. * `[3,

2, 5]` whose elements are **not** consecutive.

Example 2:

Input: nums = [2,2,2,2,2], k = 4

Output: [-1,-1]

Example 3:

Input: nums = [3,2,3,2,3,2], k = 2

Output: [-1,3,-1,3,-1]

Constraints:

* `1 <= n == nums.length <= 105` * `1 <= nums[i] <= 106` * `1 <= k <= n`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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