Problem 2382: Maximum Segment Sum After Removals

Problem Information

Difficulty: Hard
Acceptance Rate: 49.14%
Paid Only: No
Tags: Array, Union Find, Prefix Sum, Ordered Set

Problem Description

You are given two **0-indexed** integer arrays `nums` and `removeQueries`, both of length `n`. For the `ith` query, the element in `nums` at the index `removeQueries[i]` is removed, splitting `nums` into different segments.

A **segment** is a contiguous sequence of **positive** integers in `nums`. A **segment sum** is the sum of every element in a segment.

Return _an integer array_`answer` _, of length_`n` _, where_`answer[i]`_is the**maximum** segment sum after applying the _`ith` _removal._

Note: The same index will **not** be removed more than once.

Example 1:

Input: nums = [1,2,5,6,1], removeQueries = [0,3,2,4,1] **Output:** [14,7,2,2,0] **Explanation:** Using 0 to indicate a removed element, the answer is as follows: Query 1: Remove the 0th element, nums becomes [0,2,5,6,1] and the maximum segment sum is 14 for segment [2,5,6,1]. Query 2: Remove the 3rd element, nums becomes [0,2,5,0,1] and the maximum segment sum is 7 for segment [2,5]. Query 3: Remove the 2nd element, nums becomes [0,2,0,0,1] and the maximum segment sum is 2 for segment [2]. Query 4: Remove the 4th element, nums becomes [0,2,0,0,0] and the maximum segment sum is 2 for segment [2]. Query 5: Remove the 1st element, nums becomes [0,0,0,0,0] and the maximum segment sum is 0, since there are no segments. Finally, we return [14,7,2,2,0].

Example 2:

Input: nums = [3,2,11,1], removeQueries = [3,2,1,0] **Output:** [16,5,3,0] **Explanation:** Using 0 to indicate a removed element, the answer is as follows: Query 1: Remove the 3rd element, nums becomes [3,2,11,0] and the maximum segment sum is 16 for segment [3,2,11]. Query 2: Remove the 2nd element, nums becomes [3,2,0,0] and the maximum segment sum is 5 for segment [3,2]. Query 3: Remove the 1st element, nums becomes [3,0,0,0] and the maximum segment sum is 3 for segment [3]. Query 4: Remove the 0th element, nums becomes [0,0,0,0] and the maximum segment sum is 0, since there are no segments. Finally, we return [16,5,3,0].

Constraints:

* `n == nums.length == removeQueries.length` * `1 <= n <= 105` * `1 <= nums[i] <= 109` * `0 <= removeQueries[i] < n` * All the values of `removeQueries` are **unique**.

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public long[] maximumSegmentSum(int[] nums, int[] removeQueries) {

}
}
```

Python3:

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