2382. Maximum Segment Sum After Removals

My Submissions (/contest/biweekly-contest-85/problems/maximum-segment-sum-after-removals/submissions/)

Back to Contest (/contest/biweekly-contest-85/)

You are given two **0-indexed** integer arrays `nums` and `removeQueries`, both of length `n`. For the i^{th} 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* i^{th} *removal.*

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

User Accepted:	802
User Tried:	1770
Total Accepted:	868
Total Submissions:	3197
Difficulty:	Hard

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 <= 10^5`
- `1 <= nums[i] <= 10^9`
- `0 <= removeQueries[i] < n`
- All the values of `removeQueries` are **unique**.

Discuss (https://leetcode.com/problems/maximum-segment-sum-after-removals/discuss)

JavaScript

```
function DJSet(n) {
    let p = Array(n).fill(-1), s = Array(n).fill(0); // sz: group prefix sum
    return { find, union, update, sum, par }
    function find(x) {
        return p[x] < 0 ? x : p[x] = find(p[x]);
    }
    function union(x, y) {
        x = find(x);
        y = find(y);
        if (x == y) return false;
        if (p[x] < p[y]) [x, y] = [y, x];
        p[x] += p[y];
```

```
        p[y] = x;
        s[x] += s[y];
        return true;
    }
    function update(idx, v) {
        s[idx] = v;
        // s[idx] += v;
    }
    function sum() {
        return s;
    }
    function par() {
        return p;
    }
}

const maximumSegmentSum = (a, b) => {
    let n = a.length, res = [0], ds = new DJSet(n), used = new Set(), max = 0;
    for (let i = n - 1; i >= 1; i--) {
        used.add(b[i]);
        ds.update(b[i], a[b[i]]);
        if (used.has(b[i] - 1)) ds.union(b[i], b[i] - 1);
        if (used.has(b[i] + 1)) ds.union(b[i], b[i] + 1);
        max = Math.max(max, ds.sum()[ds.find([b[i]])]);
        res.push(max);
    }
    return res.reverse();
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/1029203822/) | More Details > (/submissions/detail/1029203822/)

Share your acceptance!