

6351. Find Score of an Array After Marking All Elements

My Submissions (/contest/biweekly-contest-100/problems/find-score-of-an-array-after-marking-all-elements/submissions/)

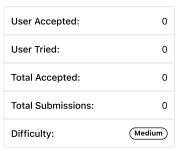
Back to Contest (/contest/biweekly-contest-100/)

You are given an array nums consisting of positive integers.

Starting with score = 0, apply the following algorithm:

- Choose the smallest integer of the array that is not marked. If there is a tie, choose the one with the smallest index.
- Add the value of the chosen integer to score.
- Mark the chosen element and its two adjacent elements if they exist.
- Repeat until all the array elements are marked.

Return the score you get after applying the above algorithm.



Example 1:

```
Input: nums = [2,1,3,4,5,2]
Output: 7
Explanation: We mark the elements as follows:
- 1 is the smallest unmarked element, so we mark it and its two adjacent elements: [2,1,3,4,5,2].
- 2 is the smallest unmarked element, so we mark it and its left adjacent element: [2,1,3,4,5,2].
- 4 is the only remaining unmarked element, so we mark it: [2,1,3,4,5,2].
Our score is 1 + 2 + 4 = 7.
```

Example 2:

```
Input: nums = [2,3,5,1,3,2]
Output: 5
Explanation: We mark the elements as follows:
- 1 is the smallest unmarked element, so we mark it and its two adjacent elements: [2,3,5,1,3,2].
- 2 is the smallest unmarked element, since there are two of them, we choose the left-most one, so we mark the one at index 0 a - 2 is the only remaining unmarked element, so we mark it: [2,3,5,1,3,2].
Our score is 1 + 2 + 2 = 5.
```

Constraints:

- 1 <= nums.length <= 10⁵
- $1 \le nums[i] \le 10^6$

```
Java
                                                                                                                       Ø
                                                                                                                             \mathfrak{C}
1
    // package leetcode.biweekly.r100;
2
3 ▼
   import java.util.*;
4
5 ▼
   public class Solution {
        public long findScore(int□ a) {
6
7
            int n = a.length;
8
            long res = 0;
9
            TreeMap<Integer, TreeSet<Integer>> m = counter_value_in_indexA_in(a);
            while (m.size() > 0) {
10
11
                 int x = m.firstKey();
12
                res += x;
13
                TreeSet<Integer> ia = m.get(x), iar = null, ial = null;
14
                int pickIdx = ia.first();
15
                Integer l = null, r = null;
16
                if (pickIdx + 1 < n) {
17
                     r = a[pickIdx + 1];
18
                     iar = m.get(r);
19
20 •
                if (pickIdx - 1 >= 0) {
                     l = a[pickIdx - 1];
```

```
ial = m.get(l);
22
23
                }
                 // tr(x, l, r, m);
24
25
                if (l != null) op(m, ial, pickIdx - 1, l);
26
                if (r != null) op(m, iar, pickIdx + 1, r);
27
                op(m, ia, pickIdx, x);
28
                 // tr("remove", m);
29
30
            return res;
        }
31
32
33 ▼
        void op(TreeMap<Integer, TreeSet<Integer>>> m, TreeSet<Integer> ia, int idx, int v) {
34 ▼
            if (ia != null) {
                 ia.remove(idx);
35
36 ▼
                 if (ia.size() > 0) {
37
                     m.put(v, ia);
38 ▼
                } else {
39
                     m.remove(v);
40
41
            }
42
        }
43
        TreeMap<Integer, TreeSet<Integer>> counter_value_in_indexA_in(int[] a) {
44 🔻
            TreeMap<Integer, TreeSet<Integer>> m = new TreeMap<>();
45
46 ▼
            for (int i = 0; i < a.length; i++) {
                 if (!m.containsKey(a[i])) m.put(a[i], new TreeSet⇔());
47
48
                m.get(a[i]).add(i);
49
50
            return m;
51
        }
52
    }
53
```

☐ Custom Testcase

Use Example Testcases

Submission Result: Accepted (/submissions/detail/917593508/) @

More Details > (/submissions/detail/917593508/)

Run

Share your acceptance!

∢1

Copyright @ 2023 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)