

6190. Find All Good Indices

My Submissions (/contest/weekly-contest-312/problems/find-all-good-indices/submissions/)

Back to Contest (/contest/weekly-contest-312/)

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

We call an index `i` in the range `k <= i < n - k` **good** if the following conditions are satisfied:

- The `k` elements that are just **before** the index `i` are in **non-increasing** order.
- The `k` elements that are just **after** the index `i` are in **non-decreasing** order.

Return an array of all good indices sorted in **increasing** order.

User Accepted: 0

User Tried: 0

Total Accepted: 0

Total Submissions: 0

Difficulty: Medium

Example 1:

Input: `nums = [2,1,1,1,3,4,1]`, `k = 2`
Output: `[2,3]`
Explanation: There are two good indices in the array:
- Index 2. The subarray `[2,1]` is in non-increasing order, and the subarray `[1,3]` is in non-decreasing order.
- Index 3. The subarray `[1,1]` is in non-increasing order, and the subarray `[3,4]` is in non-decreasing order.
Note that the index 4 is not good because `[4,1]` is not non-decreasing.

Example 2:

Input: `nums = [2,1,1,2]`, `k = 2`
Output: `[]`
Explanation: There are no good indices in this array.

Constraints:

- `n == nums.length`
- `3 <= n <= 105`
- `1 <= nums[i] <= 106`
- `1 <= k <= n / 2`

JavaScript

db

↺

⚙

```
1 const cutMaxConsecutiveWithIndex_de = (a) => {
2   let d = [], ia = [], l = 0, n = a.length;
3   for (let i = 0; i + 1 < n; i++) {
4     if (a[i + 1] > a[i]) {
5       d.push(a.slice(l, i + 1));
6       ia.push([l, i]);
7       l = i + 1;
8     }
9   }
10  d.push(a.slice(l));
11  ia.push([l, n - 1]);
12  return [d, ia];
13 };
14
15 const cutMaxConsecutiveWithIndex_in = (a) => {
16   let d = [], ia = [], l = 0, n = a.length;
17   for (let i = 0; i + 1 < n; i++) {
18     if (a[i + 1] < a[i]) {
19       d.push(a.slice(l, i + 1));
20       ia.push([l, i]);
21       l = i + 1;
22     }
23   }
24  d.push(a.slice(l));
25  ia.push([l, n - 1]);
26  return [d, ia];
27 };
28
29 const goodIndices = (a, k) => {
30   let [inc, inc_ia] = cutMaxConsecutiveWithIndex_in(a), [dec, dec_ia] = cutMaxConsecutiveWithIndex_de(a)
```

```

31 // pr("inc", inc, inc_ia)
32 // pr("dec", dec, dec_ia)
33 let n = a.length, res = [];
34 for (let i = 0; i < n; i++) {
35   if (i >= k && i < n - k) {
36     let bl = i - k, br = i - 1, al = i + 1, ar = i + k;
37     // let b = a.slice(bl, br + 1), c = a.slice(al, ar + 1);
38     // pr("\nindex", i);
39     // pr("before", b, "after", c)
40     // pr("beforeRange", [bl, br], "afterRange", [al, ar])
41     let beforeOK = false, afterOK = false;
42     if (valid(bl, br, n) && cover(bl, br, dec_ia)) {
43       // pr("before ok")
44       beforeOK = true;
45     }
46     if (valid(al, ar, n) && cover(al, ar, inc_ia)) {
47       // pr("after ok")
48       afterOK = true;
49     }
50     if (beforeOK && afterOK) res.push(i);
51   }
52 }
53 return res;
54 };
55
56 const cover = (l, r, a) => {
57   for (const [start, end] of a) {
58     if (l >= start && r <= end) {
59       // pr([l, r], [start, end])
60       return true;
61     }
62   }
63   return false;
64 };
65
66 const valid = (l, r, n) => l >= 0 && r < n;


```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)Submission Result: **Accepted** (/submissions/detail/807979668/) ⓘ[More Details > \(/submissions/detail/807979668/\)](#)

Share your acceptance!

◀ 1

Copyright © 2022 LeetCode

[Help Center \(/support\)](#) | [Jobs \(/jobs\)](#) | [Bug Bounty \(/bugbounty\)](#) | [Online Interview \(/interview/\)](#) | [Students \(/student\)](#) | [Terms \(/terms\)](#) | [Privacy Policy \(/privacy\)](#) [United States \(/region\)](#)