845. Longest Mountain in Array

My Submissions (/contest/weekly-contest-87/problems/longest-mountain-in-array/submissions/) Back to Contest (/contest/weekly-contest-87/)

User Accepted:	1154
User Tried:	1384
Total Accepted:	1170
Total Submissions:	4306
Difficulty:	Medium

You may recall that an array `arr` is a **mountain array** if and only if:

- `arr.length >= 3`
- There exists some index `i` (**0-indexed**) with `0 < i < arr.length - 1` such that:
  - `arr[0] < arr[1] < ... < arr[i - 1] < arr[i]`
  - `arr[i] > arr[i + 1] > ... > arr[arr.length - 1]`

Given an integer array `arr`, return *the length of the longest subarray, which is a mountain*. Return `0` if there is no mountain subarray.

Example 1:

```
Input: arr = [2,1,4,7,3,2,5]
Output: 5
Explanation: The largest mountain is [1,4,7,3,2] which has length 5.
```

Example 2:

```
Input: arr = [2,2,2]
Output: 0
Explanation: There is no mountain.
```

Constraints:

- $1 <= arr.length <= 10^4$
- $0 <= arr[i] <= 10^4$

Follow up:

- Can you solve it using only one pass?
- Can you solve it in `O(1)` space?

Discuss (https://leetcode.com/problems/longest-mountain-in-array/discuss)

JavaScript

```
const cutMaxConsecutiveWithIndex_de = (a) => {
    let d = [], ia = [], l = 0, n = a.length;
    for (let i = 0; i + 1 < n; i++) {
        if (a[i + 1] >= a[i]) {
            d.push(a.slice(l, i + 1));
            ia.push([l, i]);
            l = i + 1;
        }
    }
    d.push(a.slice(l));
    ia.push([l, n - 1]);
    return [d, ia];
};

const cutMaxConsecutiveWithIndex_in = (a) => {
    let d = [], ia = [], l = 0, n = a.length;
    for (let i = 0; i + 1 < n; i++) {
        if (a[i + 1] <= a[i]) {
            d.push(a.slice(l, i + 1));
            ia.push([l, i]);
            l = i + 1;
        }
    }
    d.push(a.slice(l));
    ia.push([l, n - 1]);
    return [d, ia];
};

const longestMountain = (a) => {
    let [inc, inc_ia] = cutMaxConsecutiveWithIndex_in(a), [dec, dec_ia] = cutMaxConsecutiveWithIndex_de(a)
    // pr("inc", inc)
    // pr("inc_ia", inc_ia)
    // pr("dec", dec)
    // pr("dec_ia", dec_ia)
    let res = 0;
    for (const [l, r] of inc_ia) {
        let incLen = cal(l, r);
        // pr("\n", [l, r])
        if (l == r) continue;
        for (const [L, R] of dec_ia) {
            if (L == R) continue;
            if (a[r + 1] < a[r] && r + 1 >= L && r + 1 <= R) {
                // pr([L, R])
                let decLen = cal(r + 1, R), len = incLen + decLen;
                // pr('len', incLen, decLen)
                res = Math.max(res, len);
                break;
            }

        }
    }
    return res;
};

const cal = (l, r) => r - l + 1;
```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/835896197/) More Details (/submissions/detail/835896197/)

Share your acceptance!

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

United States (/region)