5559. Minimum Number of Removals to Make Mountain Array

My Submissions (/contest/biweekly-contest-40/problems/minimum-number-of-removals-to-make-mountain-array/submissions/)

Back to Contest (/contest/biweekly-contest-40/)

User Accepted:	0
User Tried:	2
Total Accepted:	0
Total Submissions:	2
Difficulty:	Hard

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 `nums`, return *the **minimum** number of elements to remove to make* `nums` *a **mountain array***.

Example 1:

```
Input: nums = [1,3,1]
Output: 0
Explanation: The array itself is a mountain array so we do not need to remove any elements.
```

Example 2:

```
Input: nums = [2,1,1,5,6,2,3,1]
Output: 3
Explanation: One solution is to remove the elements at indices 0, 1, and 5, making the array nums = [1,5,6,3,1].
```

Example 3:

```
Input: nums = [4,3,2,1,1,2,3,1]
Output: 4
```

Example 4:

```
Input: nums = [1,2,3,4,4,3,2,1]
Output: 1
```

Constraints:

- `3 <= nums.length <= 1000`
- `1 <= nums[i] <= 10^9`
- It is guaranteed that you can make a mountain array out of `nums`.

JavaScript

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var minimumMountainRemovals = function(nums) {

};
```