

2216. Minimum Deletions to Make Array Beautiful

Description

You are given a **0-indexed** integer array `nums`. The array `nums` is **beautiful** if:

- `nums.length` is even.
- `nums[i] != nums[i + 1]` for all `i % 2 == 0`.

Note that an empty array is considered beautiful.

You can delete any number of elements from `nums`. When you delete an element, all the elements to the right of the deleted element will be **shifted one unit to the left** to fill the gap created and all the elements to the left of the deleted element will remain **unchanged**.

Return *the minimum number of elements to delete from `nums` to make it beautiful*.

Example 1:

Input: `nums = [1,1,2,3,5]`

Output: 1

Explanation: You can delete either `nums[0]` or `nums[1]` to make `nums = [1,2,3,5]` which is beautiful. It can be proven you need at least 1 deletion to make `nums` beautiful.

Example 2:

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

Output: 2

Explanation: You can delete `nums[0]` and `nums[5]` to make `nums = [1,2,2,3]` which is beautiful. It can be proven you need at least 2 deletions to make `nums` beautiful.

Constraints:

- `1 <= nums.length <= 105`
- `0 <= nums[i] <= 105`

