Problem 2216: Minimum Deletions to Make Array Beautiful

Problem Information

Difficulty: Medium
Acceptance Rate: 49.49%
Paid Only: No
Tags: Array, Stack, Greedy

Problem 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`

Code Snippets

C++:

```
class Solution {
public:
int minDeletion(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int minDeletion(int[] nums) {

}
}
```

Python3:

```
class Solution:
def minDeletion(self, nums: List[int]) -> int:
```