

6005. Minimum Operations to Make the Array Alternating

My Submissions (/contest/weekly-contest-280/problems/minimum-operations-to-make-the-array-alternating/submissions/)

Back to Contest (/contest/weekly-contest-280/)

You are given a **0-indexed** array `nums` consisting of `n` positive integers.

The array `nums` is called **alternating** if:

- `nums[i - 2] == nums[i]`, where $2 \leq i \leq n - 1$.
- `nums[i - 1] != nums[i]`, where $1 \leq i \leq n - 1$.

In one **operation**, you can choose an index `i` and **change** `nums[i]` into **any** positive integer.

Return the **minimum number of operations** required to make the array alternating.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

Input: `nums = [3,1,3,2,4,3]`
Output: 3
Explanation:
One way to make the array alternating is by converting it to `[3,1,3,1,3,1]`.
The number of operations required in this case is 3.
It can be proven that it is not possible to make the array alternating in less than 3 operations.

Example 2:

Input: `nums = [1,2,2,2,2]`
Output: 2
Explanation:
One way to make the array alternating is by converting it to `[1,2,1,2,1]`.
The number of operations required in this case is 2.
Note that the array cannot be converted to `[2,2,2,2,2]` because in this case `nums[0] == nums[1]` which violates the conditions.


Constraints:

- $1 \leq \text{nums.length} \leq 10^5$
- $1 \leq \text{nums}[i] \leq 10^5$

JavaScript

```
1 const counter = (a_or_s) => { let m = new Map(); for (const x of a_or_s) m.set(x, m.get(x) + 1 || 1); return m; };
2 const stmvalue_de = (m) => new Map([...m].sort((x, y) => y[1] - x[1]));
3
4 const minimumOperations = (a) => {
5   let n = a.length, even = [], odd = [], res = 0;
6   for (let i = 0; i < n; i++) i & 1 ? even.push(a[i]) : odd.push(a[i]);
7   // pr(odd)
8   // pr(even)
9   let m1 = counter(odd), m2 = counter(even);
10  m1 = stmvalue_de(m1);
11  m2 = stmvalue_de(m2);
12  let v1 = m1.keys().next().value, v2 = m2.keys().next().value;
13  // pr(m1, m2, v1, v2);
14  if (v1 !== v2) {
15    res = cal(odd, even, v1, v2);
16  } else {
17    let pre = v2;
18    v2 = -1;
19    for (const [x, ] of m2) {
20      if (x !== v1) {
21        v2 = x;
```


```
22         break;
23     }
24 }
25 let res1 = cal(odd, even, v1, v2);
26 v2 = pre;
27 v1 = -1;
28 for (const [x, ] of m1) {
29     if (x !== v2) {
30         v1 = x;
31         break;
32     }
33 }
34 let res2 = cal(odd, even, v1, v2);
35 // pr(res1, res2);
36 res = Math.min(res1, res2);
37 }
38 return res;
39 };
40
41 const cal = (odd, even, v1, v2) => {
42     let res = 0;
43     for (const e of odd) {
44         if (e !== v1) res++;
45     }
46     for (const e of even) {
47         if (e !== v2) res++;
48     }
49     return res;
50 };
```

☐ Custom Testcase☒ Use Example Testcases Run SubmitSubmission Result: **Accepted** (/submissions/detail/640252335/) ?

More Details > (/submissions/detail/640252335/)

Share your acceptance!

Copyright © 2022 LeetCode

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