

## 5977. Minimum Swaps to Group All 1's Together II

[My Submissions \(/contest/weekly-contest-275/problems/minimum-swaps-to-group-all-1s-together-ii/submissions/\)](/contest/weekly-contest-275/problems/minimum-swaps-to-group-all-1s-together-ii/submissions/)
[Back to Contest \(/contest/weekly-contest-275/\)](/contest/weekly-contest-275/)

A **swap** is defined as taking two **distinct** positions in an array and swapping the values in them.

A **circular** array is defined as an array where we consider the **first** element and the **last** element to be **adjacent**.

Given a **binary circular** array `nums`, return the *minimum number of swaps required to group all 1's present in the array together at **any** location*.

### Example 1:

**Input:** `nums = [0,1,0,1,1,0,0]`  
**Output:** 1  
**Explanation:** Here are a few of the ways to group all the 1's together:  
`[0,0,1,1,1,0,0]` using 1 swap.  
`[0,1,1,1,0,0,0]` using 1 swap.  
`[1,1,0,0,0,0,1]` using 2 swaps (using the circular property of the array).  
 There is no way to group all 1's together with 0 swaps.  
 Thus, the minimum number of swaps required is 1.

### Example 2:

**Input:** `nums = [0,1,1,1,0,0,1,1,0]`  
**Output:** 2  
**Explanation:** Here are a few of the ways to group all the 1's together:  
`[1,1,1,0,0,0,0,1,1]` using 2 swaps (using the circular property of the array).  
`[1,1,1,1,1,0,0,0,0]` using 2 swaps.  
 There is no way to group all 1's together with 0 or 1 swaps.  
 Thus, the minimum number of swaps required is 2.

### Example 3:

**Input:** `nums = [1,1,0,0,1]`  
**Output:** 0  
**Explanation:** All the 1's are already grouped together due to the circular property of the array.  
 Thus, the minimum number of swaps required is 0.

### Constraints:

- $1 \leq \text{nums.length} \leq 10^5$
- `nums[i]` is either 0 or 1.

JavaScript



```

1 const minSwaps = (a) => {
2   let res1 = groupTogether(a, 1);
3   let res0 = groupTogether(a, 0);
4   // pr(res1, res0);
5   return Math.min(res1, res0);
6 };
7
8 const groupTogether = (a, c) => {
9   let n = a.length, cntC = 0;
10  for (let i = 0; i < n; i++) {
11    if (a[i] == c) cntC++;
12  }
13  let x = cntC, max = Number.MIN_VALUE, preCompute = Array(n).fill(0);


```

```
14     if (a[0] == c) preCompute[0] = 1;
15     for (let i = 1; i < n; i++) {
16         if (a[i] == c) {
17             preCompute[i] = preCompute[i - 1] + 1;
18         } else {
19             preCompute[i] = preCompute[i - 1];
20         }
21     }
22     for (let i = x - 1; i < n; i++) {
23         if (i == x - 1) {
24             cntC = preCompute[i];
25         } else {
26             cntC = preCompute[i] - preCompute[i - x];
27         }
28         if (max < cntC) max = cntC;
29     }
30     return Math.max(x - max, 0);
31 };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)**Submission Result: Accepted** (/submissions/detail/615987465/) ?[More Details](#) (/submissions/detail/615987465/)

Share your acceptance!

Copyright © 2022 LeetCode

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