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 <= nums.length <= 105
* nums[i] is either 0 or 1.