Given a binary array data, return the minimum number of swaps required to group all 1’s present in the array together in **any place** in the array.

**Example 1:**

**Input:** data = [1,0,1,0,1]

**Output:** 1

**Explanation:**

There are 3 ways to group all 1's together:

[1,1,1,0,0] using 1 swap.

[0,1,1,1,0] using 2 swaps.

[0,0,1,1,1] using 1 swap.

The minimum is 1.

**Example 2:**

**Input:** data = [0,0,0,1,0]

**Output:** 0

**Explanation:**

Since there is only one 1 in the array, no swaps needed.

**Example 3:**

**Input:** data = [1,0,1,0,1,0,0,1,1,0,1]

**Output:** 3

**Explanation:**

One possible solution that uses 3 swaps is [0,0,0,0,0,1,1,1,1,1,1].

**Example 4:**

**Input:** data = [1,0,1,0,1,0,1,1,1,0,1,0,0,1,1,1,0,0,1,1,1,0,1,0,1,1,0,0,0,1,1,1,1,0,0,1]

**Output:** 8

**Constraints:**

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