Problem 3191: Minimum Operations to Make Binary Array Elements Equal to One I

Problem Information

Difficulty: Medium
Acceptance Rate: 80.46%
Paid Only: No
Tags: Array, Bit Manipulation, Queue, Sliding Window, Prefix Sum

Problem Description

You are given a binary array `nums`.

You can do the following operation on the array **any** number of times (possibly zero):

* Choose **any** 3 **consecutive** elements from the array and **flip** **all** of them.

Flipping an element means changing its value from 0 to 1, and from 1 to 0.

Return the **minimum** number of operations required to make all elements in `nums` equal to 1. If it is impossible, return -1.

Example 1:

Input: nums = [0,1,1,1,0,0]

Output: 3

Explanation: We can do the following operations:

* Choose the elements at indices 0, 1 and 2. The resulting array is `nums = [_**1**_ ,_**0**_ ,_**0**_ ,1,0,0]`. * Choose the elements at indices 1, 2 and 3. The resulting array is `nums = [1,_**1**_ ,_**1**_ ,**_0_** ,0,0]`. * Choose the elements at indices 3, 4 and 5. The resulting array is `nums = [1,1,1,**_1_** ,_**1**_ ,_**1**_]`.

Example 2:

Input: nums = [0,1,1,1]

Output: -1

Explanation: It is impossible to make all elements equal to 1.

Constraints:

* `3 <= nums.length <= 105` * `0 <= nums[i] <= 1`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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