Problem 487: Max Consecutive Ones II

Problem Information

Difficulty: Medium
Acceptance Rate: 51.79%
Paid Only: Yes
Tags: Array, Dynamic Programming, Sliding Window

Problem Description

Given a binary array `nums`, return _the maximum number of consecutive_`1` _' s in the array if you can flip at most one_ `0`.

Example 1:

Input: nums = [1,0,1,1,0] **Output:** 4 **Explanation:** - If we flip the first zero, nums becomes [1,1,1,1,0] and we have 4 consecutive ones. - If we flip the second zero, nums becomes [1,0,1,1,1] and we have 3 consecutive ones. The max number of consecutive ones is 4.

Example 2:

Input: nums = [1,0,1,1,0,1] **Output:** 4 **Explanation:** - If we flip the first zero, nums becomes [1,1,1,1,0,1] and we have 4 consecutive ones. - If we flip the second zero, nums becomes [1,0,1,1,1,1] and we have 4 consecutive ones. The max number of consecutive ones is 4.

Constraints:

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

Follow up: What if the input numbers come in one by one as an infinite stream? In other words, you can't store all numbers coming from the stream as it's too large to hold in memory. Could you solve it efficiently?

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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