75 Days of Code

Day17 Problem no: 1493. Longest Subarray of 1's After Deleting One Element (leetcode)

Given a binary array nums, you should delete one element from it.

Return the size of the longest non-empty subarray containing only 1's in the resulting array. Return 0 if there is no such subarray.

Example 1:

Input: nums = [1,1,0,1]
Output: 3
Explanation: After deleting the number in position 2, [1,1,1] contains 3 numbers with value of 1's.
Example 2:

Input: nums = [0,1,1,1,0,1,1,0,1]
Output: 5
Explanation: After deleting the number in position 4, [0,1,1,1,1,1,0,1] longest subarray with value of 1's is [1,1,1,1,1].
Example 3:

Input: nums = [1,1,1]
Output: 2
Explanation: You must delete one element.

Bolded numbers were flipped from 0 to 1. The longest subarray is underlined.

For this problem , we are gonna use sliding window approach

Key terms to know problem can be solved by sliding window :

Question includes : **Array ,SubArray , SubString , Largest , Smallest ,Maximum and Minimum with window size may or may not present**

Solution of the above problem using sliding window

1. Initialize two pointers as start and zeroCount .
2. Start the loop , when current number is zero then decrease the zeroConter and keep the window size increasing
3. If zeroCounter becomes negative then , start a loop to reduce the window size by removing the element up to zero element's index , while increasing startPointer , when it finds the zero element in the window then increase zeroCounter
4. At each iteration compare the maximum from index to where the start pointer is to get largest subarray of 1

```
function longestSubarray(nums: number[]): number {
  let startPointer = 0;
  let zeroCount = 1;
  let maxOnes = Number.MIN_VALUE;
  for (let numIndex = 0; numIndex < nums.length; numIndex++) {
    if (nums[numIndex] == 0) {
      zeroCount--;
    }

    while (zeroCount < 0) {
      if (nums[startPointer] === 0) {
        zeroCount++;
      }
      startPointer++;
    }

    maxOnes = Math.max(maxOnes, numIndex - startPointer);
  }
  return maxOnes;
}

let answer = longestSubarray([0,1,1,1,0,1,1,0,1]);
console.log("Answer :",answer);
```

```
[Running] node "c:\Users\Shubham\Desktop\75daysOfCode\75DaysOfCode\day17.js"
Answer : 5
```