1004. Max Consecutive Ones III

Given a binary array `nums` and an integer `k`, return *the maximum number of consecutive* `1`*'s in the array if you can flip at most* `k` `0`'s.

Example 1:

```
Input: nums = [1,1,1,0,0,0,1,1,1,1,0], k = 2
Output: 6
Explanation: [1,1,1,0,0,1,1,1,1,1,1]
Bolded numbers were flipped from 0 to 1. The longest subarray is
underlined.
```

Example 2:

```
Input: nums = [0,0,1,1,0,0,1,1,1,0,1,1,0,0,0,1,1,1,1], k = 3
Output: 10
Explanation: [0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1]
Bolded numbers were flipped from 0 to 1. The longest subarray is
underlined.
```

Constraints:

- `1 <= nums.length <= 10<sup>5</sup>`
- `nums[i]` is either `0` or `1`.
- `0 <= k <= nums.length`
- ```
  class Solution:
      def longestOnes(self, nums: List[int], k: int) -> int:
          ans = 0
          j = -1
          count = 0
          for i in range(len(nums)):
              if nums[i]==0:
                  count = count+1

              while count>k:
                  j = j+1
                  if nums[j]==0:
                      count = count-1
  ```
  ```

```
            ans  = max(ans,i-j)

        return ans
```