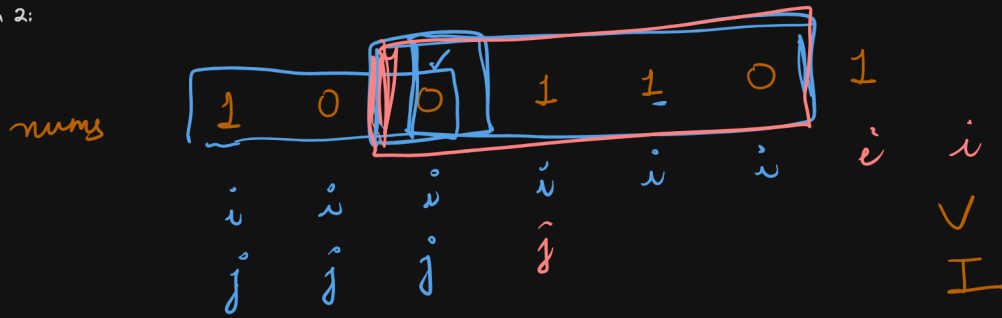




Approach 2:



$$\max(\max, i - j + 1)$$

$$\begin{aligned} \max &= 0 \times 1 \times 1 \times 1 \times 1 \times 1 \\ \text{zeros} &= 0 \times 1 \times 1 \times 1 \times 1 \times 1 \end{aligned}$$

Time complexity:  
 $O(N)$   
 Space Complexity:  
 $O(1)$

2N

```
class Solution {
    public int findMaxConsecutiveOnes(int[] nums) {
        // approach 2
        int max = 0;

        for (int i = 0; i < nums.length; i++) {
            int zeroes = 0;
            for (int j = i; j < nums.length; j++) {
                if (nums[j] == 0) {
                    zeroes += 1;
                }

                if (zeroes > 1) {
                    break;
                }

                max = Math.max(max, j - i + 1);
            }
        }

        return max;
    }
}
```

```
class Solution {
    public int findMaxConsecutiveOnes(int[] nums) {
        int i = 0,
            j = 0,
            max = 0,
            zeroes = 0;

        while (i < nums.length) {
            if (nums[i] == 0) {
                zeroes += 1;
            }
        }
    }
}
```

```
        while (zeroes == 2) {  
            if (nums[j] == 0) {  
                zeroes -= 1;  
            }  
            j++;  
        }  
  
        max = Math.max(max, i - j + 1);  
        i++;  
    }  
  
    return max;  
}
```