Max Consecutive Ones

Given a binary array, find the maximum number of consecutive 1s in this array.

Example 1:

```
Input: [1,1,0,1,1,1]
Output: 3
Explanation: The first two digits or the last three digits are consecutive 1s.
    The maximum number of consecutive 1s is 3.
```

Note:

- The input array will only contain `0` and `1`.
- The length of input array is a positive integer and will not exceed 10,000

Solution 1

```
    public int findMaxConsecutiveOnes(int[] nums) {
        int maxHere = 0, max = 0;
        for (int n : nums)
            max = Math.max(max, maxHere = n == 0 ? 0 : maxHere + 1);
        return max;
    }
```

The idea is to reset `maxHere` to 0 if we see 0, otherwise increase `maxHere` by 1
The max of all `maxHere` is the solution

```
110111
^ maxHere = 1

110111
.^ maxHere = 2

110111
..^ maxHere = 0

110111
...^ maxHere = 1

110111
....^ maxHere = 2

110111
.....^ maxHere = 3
```

We can also solve this problem by setting `k = 0` of Max Consecutive Ones II written by yuxiangmusic original link here

Solution 2

This is a really easy problem. No explanation :)

```
public class Solution {
    public int findMaxConsecutiveOnes(int[] nums) {
        int result = 0;
        int count = 0;

        for (int i = 0; i < nums.length; i++) {
            if (nums[i] == 1) {
         count++;
         result = Math.max(count, result);
            }
            else count = 0;
        }

        return result;
    }
}
```

written by shawngao original link here

Solution 3

```
class Solution(object):
    def findMaxConsecutiveOnes(self, nums):
        cnt = 0
        ans = 0
        for num in nums:
            if num == 1:
                cnt += 1
                ans = max(ans, cnt)
            else:
                cnt = 0
        return ans
```

written by Ipeq1 original link here