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

```
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

Solution 2

```java
    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
```

written by yuxiangmusic original link here

Solution 3

```c
int findMaxConsecutiveOnes(int* nums, int numsSize) {
    int i, count, max;

    max = count = 0;
    for (i = 0 ; i < numsSize ; ++i)
        if (nums[i])
            if (++count > max) max = count;
        else count = 0;

    return max;
}
```

written by cifer-lee original link here

From Leetcoder.