

3.8

Max Consecutive Ones

EASY

■ What You Will Learn

- ✓ Linear scan with a running counter
- ✓ Reset on zeros, update max on each step
- ✓ Classic sliding window warmup

■ Intuition First

You're counting how many green lights you can hit in a row without stopping. Every time you hit a red light (0), your streak resets to 0. You keep track of your longest streak ever.

■ The Core Idea

- count = current streak of consecutive 1s
- maxCount = best streak seen so far
- See 1: count++, update maxCount
- See 0: reset count=0

■ Visual Trace on [1,1,0,1,1,1]

Input

1	1	0	1	1	1
---	---	---	---	---	---

```
arr[0]=1 → count=1, max=1
arr[1]=1 → count=2, max=2
arr[2]=0 → count=0, max stays 2
arr[3]=1 → count=1, max stays 2
arr[4]=1 → count=2, max stays 2
arr[5]=1 → count=3, max=3 ✓
```

Max streak = 3 (last three) ✓

1	1	0	1	1	1
---	---	---	---	---	---

■ Java Code

```
public static int maxConsecutiveOnes(int[] arr) {
    int count = 0, maxCount = 0;
```

```
for (int x : arr) {  
    if (x == 1) {  
        count++;  
        maxCount = Math.max(maxCount, count);  
    } else {  
        count = 0; // Reset streak on 0  
    }  
}  
  
return maxCount;  
}
```

■ Complexity

Case	Time	Space	What happens?
All Cases	$O(n)$	$O(1)$	Single pass, two variables

■ Common Mistakes

- Update maxCount INSIDE the if block, not after — otherwise last element might be missed
- Don't return count — return maxCount

■ Extension — Sliding Window

- What if you can flip at most 1 zero? → Sliding window problem
- What if you can flip at most K zeros? → Classic sliding window LeetCode #1004

■ Practice Problems

[Easy] Count max consecutive ones in [0,0,1,1,1,0,0,1,1]

[Med] Max consecutive ones with at most 1 zero flip allowed

[Med] Max consecutive ones with at most K flips (LeetCode #1004)

■ Quick Revision

- count = running streak, maxCount = best streak
- See 1: count++, update max. See 0: count=0
- $O(n)$ time, $O(1)$ space
- Foundation for sliding window problems