Single Number II

When numbers appear three times except one, the simple XOR trick isn't enough - but we can solve it with bitwise counting.

Idea 1: Bitwise counting (conceptual):

- For each bit position (0-31), count how many numbers have that bit set.
- Since all numbers except one appear 3 times, the count for each bit must be a multiple of 3 except for the bits of the unique number.
- Take count % 3 for each bit → reconstruct the unique number.

This is $O(32 \cdot n) = O(n)$ and uses $O(1)$ extra space.

Idea 2: Bitmask finite state machine (clever $O(1)$):

We can track counts using two masks (ones, twos):

- ones: bits which have appeared exactly once so far.
- twos: bits which have appeared exactly twice so far.
- When a bit appears the third time, it gets cleared from both.

Transition rules:

```
ones = (ones ^ num) & ~twos;
twos = (twos ^ num) & ~ones;
```

At the end ones hold the unique number.

Complexity:

- Time: $O(n)$
- Space: $O(1)$
- Works with negatives too (since bitmasking handles sign bits).