

3.9 Single Number

EASY

■ What You Will Learn

- ✓ The XOR trick — most elegant bit manipulation
- ✓ Why $a \oplus a = 0$ and $a \oplus 0 = a$
- ✓ How to find the odd-one-out in $O(n)$ time and $O(1)$ space

■ Intuition First

Every number except one appears exactly twice. XOR has a magic property: a number XOR'd with itself is 0, and 0 XOR'd with any number is that number. So if you XOR everything together, all pairs cancel out and the single number survives.

■ XOR Magic Rules

- $a \oplus a = 0$ (same numbers cancel)
- $a \oplus 0 = a$ (zero is identity)
- XOR all elements $\rightarrow$ pairs cancel $\rightarrow$ single number left
- Order doesn't matter — XOR is commutative and associative

■ Visual Trace on [4,1,2,1,2]

4 appears once — all others appear twice

4	1	2	1	2
---	---	---	---	---

```
Start: xor = 0
xor ^ 4 = 4
xor ^ 1 = 4^1 = 5
xor ^ 2 = 5^2 = 7
xor ^ 1 = 7^1 = 6 (1 cancels with earlier 1)
xor ^ 2 = 6^2 = 4 (2 cancels with earlier 2)
Result = 4 ✓
```

■ Java Code

```
public static int singleNumber(int[] arr) {
    int xor = 0;
    for (int x : arr) {
        xor ^= x; // Pairs cancel out, single survives
    }
    return xor;
}
```

```

}

return xor;

}

```

■ Complexity

Case	Time	Space	What happens?
All Cases	$O(n)$	$O(1)$	One XOR loop — nothing simpler

■ Why not HashMap?

- HashMap approach works but uses $O(n)$ space
- XOR achieves same result in $O(1)$ space — always prefer XOR here
- HashSet approach: add on first encounter, remove on second, leftover is answer

■ XOR in DSA — Remember This

- $a \oplus b \oplus a = b$ (the a's cancel)
- XOR of all numbers $1..N$ has a pattern too
- This pattern extends to: find two singles (where two numbers appear once)

■ Practice Problems

[Easy] Find single number in [2,2,1]

[Med] Find two non-repeating elements in array (both appear once)

[Hard] Find number appearing once when all others appear 3 times

■ Quick Revision

- XOR all elements: pairs vanish, single survives
- $a \oplus a = 0$, $a \oplus 0 = a$ — memorize these
- $O(n)$ time, $O(1)$ space — beats HashMap
- Works because XOR is commutative and associative