

Single Number

Bit Manipulation • Easy (XOR Self-Cancellation)

The Hook

Every number in an array appears twice — except one lonely number. Find it. You cannot use extra space, and your solution must be $O(n)$. A hash map feels obvious. But there is a trick so elegant it feels like magic: XOR every element together. Pairs cancel. The survivor is your answer. This is one of the most iconic bit tricks in competitive programming.

XOR Properties — The Foundation

- $a \oplus a = 0$ (any number XORed with itself is 0)
- $a \oplus 0 = a$ (any number XORed with 0 is itself)
- XOR is commutative: $a \oplus b = b \oplus a$
- XOR is associative: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

Because of these properties, when you XOR all elements, every number that appears twice cancels ($x \oplus x = 0$). The single number remains because $x \oplus 0 = x$.

Dry Run — [4, 1, 2, 1, 2]

- $\text{result} = 0$
- $0 \oplus 4 = 4$
- $4 \oplus 1 = 5$
- $5 \oplus 2 = 7$
- $7 \oplus 1 = 6$
- $6 \oplus 2 = 4 \leftarrow \text{answer: } 4 \checkmark$

Java Implementation

```
public class SingleNumber {
    public static int singleNumber(int[] nums) {
        int result = 0;
        for (int num : nums) {
            result ^= num; // pairs cancel, survivor remains
        }
    }
}
```

```

}
return result;
}
}

```

■ Complexity

Case	Time	Space	Why
All cases	$O(n)$	$O(1)$	Single pass, only one integer variable for state

■ Common Mistakes

- Using addition/subtraction: $2 * (\text{sum of unique}) - (\text{total sum})$ works mathematically but overflows for large inputs.
- Forgetting to initialize `result = 0` before XORing.

■ Patterns & Connections

- XOR swap: $a \oplus b$; $b \oplus a$; $a \oplus b$ — swaps without temp variable
- Missing number: XOR $[0..n]$ with array elements
- Two singles: find two non-repeating numbers (split by differing bit)

■ Practice Problems

Problem	Difficulty
Single Number (LeetCode 136)	Easy
Missing Number (LeetCode 268)	Easy
Single Number II (LeetCode 137)	Medium
Single Number III (LeetCode 260)	Medium

■ Quick Revision

- XOR all elements → pairs cancel → single survives
- $a \oplus a = 0$, $a \oplus 0 = a$ — these two rules are all you need
- $O(n)$ time, $O(1)$ space — optimal solution

■ Mental Model

XOR is a 'difference detector.' Two identical things cancel to nothing. Keep XORing and everything that appears twice disappears. Only the odd one out survives — like the last person standing in musical chairs.