

Number of 1 Bits

Key Idea:

You want to count how many 1s are in the binary representation of n .

Two standard ways:

1. Bit shifting loop: check each bit with $(n \& 1)$.
2. Brian Kernighan's trick: repeatedly clear the lowest set bit ($n \&= (n-1)$), counting how many times until $n == 0$.

Complexity:

· Time: $O(k)$, where k = number of set bits (≤ 32 for 32-bit integers).

· Space: $O(1)$

This is faster than always looping 32 times if the number is sparse.

Follow-up (optimize many calls):

If you call this function millions of times:

- Use a lookup table for precomputed counts of 1s in 8-bit numbers (0..255).
- Then split a 32-bit integer into 4 bytes and sum up their precomputed counts.
- This reduces per call work to 4 lookups.