

Reverse Bits

We want to flip the order of bits in a 32-bit unsigned integer.

Key Idea:

- Loop through all 32-bits.
- At each step:
 1. Shift result left ($res \ll = 1$) to make room.
 2. Take the lowest bit from n ($n \& 1$) and add it.
 3. Shift n right ($n \gg = 1$) to process the next bit.

Complexity:

- Time: $O(32) = O(1)$ (always fixed number of steps)
- Space: $O(1)$

Follow-Up Optimization (many calls):

If `reverseBits()` is called very often:

- Precompute the reverse of 8-bit chunks (0...255) in a lookup table.
- Then reverse a 32-bit integer by splitting into 4 bytes and looking each up:
$$res = reverseTable[byte0] \ll 24$$
$$| reverseTable[byte1] \ll 16$$
$$| reverseTable[byte2] \ll 8$$
$$| reverseTable[byte3];$$
- This reduces per-call cost to $O(1)$ with only 256 precomputed entries.