Reverse Bits

Problem description

The goal of this problem is to reverse the bits of a given 32-bit unsigned integer. This means that every bit in the binary representation of the integer is inverted from end to start.

Intuition

To reverse the bits of an integer, the solution takes each bit individually from the least significant bit to the most significant bit, and places that bit into the reversed position in a new number, which would then become the most significant bit of the new number. This is repeatedly done for all 32 bits of the integer.

Starting with res as 0, which will hold the reversed bits, we loop 32 times since we're dealing with 32-bit numbers. In each iteration, we:

* Check the least significant bit of n via (n & 1) which uses the bitwise AND operator to isolate the bit.
* Shift this bit to its reversed position with << (31 - i), where i is the current iteration index, thus "moving" the bit to the correct position in the reversed number.
* Use |= (bitwise OR assignment) to add this bit to the result res.
* Right shift n by 1 with n >>= 1 to move to the next bit for the next iteration.