class Solution:

def divide(self, dividend: int, divisor: int) -> int:

# Handle overflow

if dividend == -2\*\*31 and divisor == -1:

return 2\*\*31 - 1

# Determine the sign of the result

is\_negative = (dividend < 0) ^ (divisor < 0) # XOR to check if signs differ

# Work with absolute values

abs\_dividend, abs\_divisor = abs(dividend), abs(divisor)

result = 0

# Perform division using bit manipulation

while abs\_dividend >= abs\_divisor:

temp\_divisor, multiple = abs\_divisor, 1

while abs\_dividend >= (temp\_divisor << 1):

temp\_divisor <<= 1

multiple <<= 1

abs\_dividend -= temp\_divisor

result += multiple

# Apply the sign

return -result if is\_negative else result