**Bitwise Manipulation problems** on LeetCode that are often featured in interviews:

**1. Single Number Problems**

* **Single Number** (LeetCode #136):
  + Find the number that appears only once in an array where every other number appears twice. Use XOR to solve in O(n) time and O(1) space.
* **Single Number II** (LeetCode #137):
  + Find the number that appears only once in an array where every other number appears three times.
* **Single Number III** (LeetCode #260):
  + Find two numbers that appear only once in an array where every other number appears twice.

**2. XOR and Complement Problems**

* **Maximum XOR of Two Numbers in an Array** (LeetCode #421):
  + Use a Trie and bitwise operations to find the maximum XOR of two numbers in an array.
* **Complement of Base 10 Integer** (LeetCode #1009):
  + Find the bitwise complement of a non-negative integer.

**3. Counting Bits**

* **Number of 1 Bits** (LeetCode #191):
  + Count the number of 1 bits in the binary representation of a number (Hamming Weight).
* **Counting Bits** (LeetCode #338):
  + For every number i in the range [0, n], count the number of 1 bits in i and return the results as an array.
* **Hamming Distance** (LeetCode #461):
  + Calculate the number of differing bits between two integers.

**4. Subsets and Combinations**

* **Subsets** (LeetCode #78):
  + Generate all subsets of a given set using bit manipulation to represent inclusion/exclusion.
* **Gray Code** (LeetCode #89):
  + Generate the sequence of Gray Code for a given number of bits.

**5. Shifting and Masking**

* **Reverse Bits** (LeetCode #190):
  + Reverse the bits of a given 32-bit unsigned integer.
* **Shift-and-Add for Power Calculation** (LeetCode #50):
  + Use bit manipulation for fast exponentiation.

**6. Logical Operators and Optimization**

* **Power of Two** (LeetCode #231):
  + Check if a number is a power of two using bitwise operations.
* **Power of Four** (LeetCode #342):
  + Check if a number is a power of four using bitwise operations.

**7. Advanced XOR Problems**

* **Find XOR of All Numbers in Range** (LeetCode #1486):
  + Calculate the XOR of all elements in a given range [start, start + 2 \* n - 1].
* **Sum of Two Integers** (LeetCode #371):
  + Calculate the sum of two integers without using the + or - operators, using bitwise operations instead.

**8. Miscellaneous**

* **Divide Two Integers** (LeetCode #29):
  + Implement division of two integers without using multiplication, division, or mod operators.
* **Missing Number** (LeetCode #268):
  + Find the missing number in a range using XOR to cancel out duplicates.
* **Find the Difference** (LeetCode #389):
  + Given two strings, find the extra character added to the second string using XOR.

**Tips for Solving Bitwise Problems:**

1. **Understand Basic Operations**:
   * AND (&), OR (|), XOR (^), NOT (~), Left Shift (<<), Right Shift (>>).
2. **Common Tricks**:
   * Toggle a bit: x ^= (1 << i)
   * Check if a bit is set: (x & (1 << i)) != 0
   * Set a bit: x |= (1 << i)
   * Clear a bit: x &= ~(1 << i)
3. **Pattern Recognition**:
   * XOR is useful for problems with duplicate cancellation.
   * Shifting helps in problems involving power or division.