BINARY-

Bit Hacks – Part 1 (Basic)

Bit Hacks – Part 2 (Playing with k'th bit)

Bit Hacks – Part 3 (Playing with rightmost set bit of a number)

Bit Hacks – Part 4 (Playing with letters of English alphabet)

Bit Hacks – Part 5 (Find absolute value of an integer without branching)

Bit Hacks – Part 6 (Random Problems)

Brian Kernighan's Algorithm to count set bits in an integer

Compute parity of a number using lookup table

Count set bits using lookup table

Find the minimum or maximum of two integers without using branching

Multiply 16-bit integers using 8-bit multiplier

Round up to the next highest power of 2

Round up to the previous power of 2

Swap individual bits at given position in an integer

Check if given number is power of 4 or not

Reverse Bits of a given Integer

Find odd occurring element in an array in single traversal

Find two odd occurring element in an array without using any extra space

Swap two bits at given position in an integer

Add binary representation of two integers

Swap Adjacent Bits of a Number

Print all distinct Subsets of a given Set

Perform Division of two numbers without using division operator (/)

Check if adjacent bits are set in binary representation of a given number

Conditionally negate a value without branching

Find two duplicate elements in an limited range array (using XOR)

By-Anushka Gupta

Find missing number and duplicate elements in an array

Check if given number is power of 8 or not

Circular shift on binary representation of an integer by k positions

Solve given set of problems without using multiplication or division operators

Reverse Bits of an integer using lookup table

Generate binary numbers between 1 to N

Efficiently implement power function | Recursive and Iterative

Find square of a number without using multiplication and division operator | 3 methods

Generate power set of a given set

Huffman Coding

Find all odd occurring elements in an array having limited range of elements