Coding intern prep

Bit manipulation tecniques

1)reverse a bit string

unsigned int v; // input bits to be reversed

unsigned int r = v; // r will be reversed bits of v; first get LSB of v

int s = sizeof(v) \* CHAR\_BIT - 1; // extra shift needed at end

for (v >>= 1; v; v >>= 1)

{

r <<= 1;

r |= v & 1;

s--;

}

r <<= s; // shift when v's highest bits are zero

2)check if the bits group has equal or not like true for 101 1100 110011

int first\_bit = n % 2;

    int first\_count = 1;

    n = n / 2;

    while (n % 2 == first\_bit && n > 0) {

        n = n / 2;

        first\_count++;

    }

Now similarly maintain curr count and if it is not equal at any stage with first count return false

3)powers of 2 or not

V&&(v&(v-1)) first v for ensuring 0 not present

4)abs without branching

int v; // we want to find the absolute value of v

unsigned int r; // the result goes here

int const mask = v >> sizeof(int) \* CHAR\_BIT - 1;

r = (v + mask) ^ mask;

5) **Set** a bit (where n is the bit number, and 0 is the least significant bit):

unsigned char a |= (1 << n);

Example:

a 1 0 0 0 0 0 0 0

a |= (1 << 1) = 1 0 0 0 0 0 1 0

a |= (1 << 3) = 1 0 0 0 1 0 0 0

a |= (1 << 5) = 1 0 1 0 0 0 0 0

**Clear** a bit:

unsigned char b &= ~(1 << n);

6) **Toggle** a bit:

unsigned char c ^= (1 << n);

7)find right most set bit

int position = 1;

    int m = 1;

    while (!(n & m)) {

        // left shift

        m = m << 1;

        position++;

    }

8) int Add(int x, int y)

{  // Iterate till there is no carry

    while (y != 0)

    {

        // carry now contains common

        //set bits of x and y

        int carry = x & y;

        // Sum of bits of x and y where at

        //least one of the bits is not set

        x = x ^ y;

        // Carry is shifted by one so that adding

        // it to x gives the required sum

        y = carry << 1;

    }

    return x;

}

9) unsigned int countSetBits(int n)

{

    unsigned int count = 0;

    while (n)

    {

    n &= (n-1) ;

    count++;

    }

    return count;

}

10)left rotate

return (n << d)|(n >> (INT\_BITS - d));

11)int countSetBits(int n)

{

    int count = 0;

    while (n)

    {

        count += n & 1;

        n >>= 1;

    }

    return count;

}

----------------------------------------------------------------------------------------------------