


Bit Manipulations

Agenda → Study bitwise operators //

Study few tricks using bitwise operator //

Problem solving

&

Bitwise AND (&)

0	0	→	0
1	0	→	0
0	1	→	0
1	1	→	1

A & B →

0 1 0 0 1 0 1 1	→	A
0 0 0 1 0 1 0 1	→	B
<u>0 0 0 0 0 0 0 1</u>	→	<u>1</u>

Bitwise OR (|)

		1		
→	0	0	→	0
	1	0	→	1
	0	1	→	1
	1	1	→	1

$$\begin{array}{ccccccc} & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline A | B & \rightarrow & 0 & 1 & 0 & 1 & 1 & 1 \end{array}$$

Beluse Exclusive OR - XOR (\wedge)

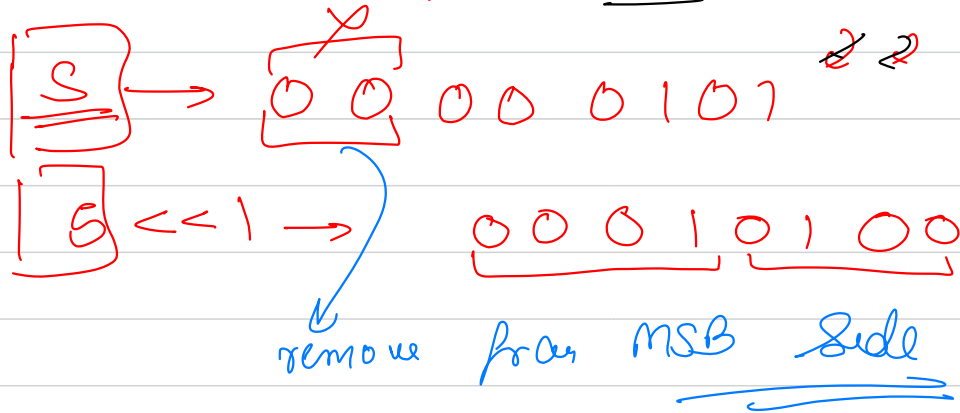
0	0	\rightarrow	0
1	0	\rightarrow	1
0	1	\rightarrow	1
1	1	\rightarrow	0

$$A \wedge B \rightarrow \begin{array}{cccccc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{array}$$

Bitwise Left Shift (\ll) \rightarrow it fills / appends new 0 bits

to the LSB side

\rightarrow You have an 8 bit number



X (0 1 0 0 1 0 1 1)

removed

X << 2

→

0 0 1 0 1 1 0 0

added

left side by appending zeros &
removing MSB

Bitwise Right Shift (\gg) \rightarrow it prepends zeros

on the MSB side, & removes LSB.

x (0 1 0 0 1 0 1) $\xrightarrow{\text{removed}}$

$x \gg 2$

$\xrightarrow{\text{added}}$
(0 0 0) 1 0 0 1 0

Bitwise Complement (\sim) \rightarrow It inverts the bits

x (0 1 0 0 1 0 1)

2nd bit

$\sim x$

\rightarrow 1 0 1 1 0 1 0 0

Q Given a number in decimal form, check if the k^{th} bit is set or not.

⇒ ① Extract the value of k^{th} bit

② Check value

Decimal ~~(X)~~

0 1 0 0 1 0 1 0

$k=2$
2nd bit

2^2

$(x \gg k)$

$x' \rightarrow$ (0 0 1 0 0 1 0)
 $1 \rightarrow$ (0 0 0 0 0 0 1)

if $((x' \gg 2) \& 1)$
else unset k^{th} bit

Q Set the k^{th} bit

Behav as? x' 0 1 0 0 1 0 1 1
 y'

2 -

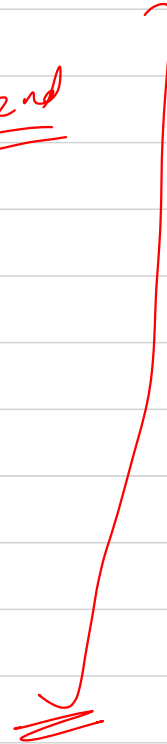
$y' = 2^{\text{nd}}$

1 \rightarrow 0 0 0 0 0 0 0 1

$y' (1 \ll 2)$ \rightarrow 0 0 0 0 0 0 1 0 0

$y' = 1 \ll 2$

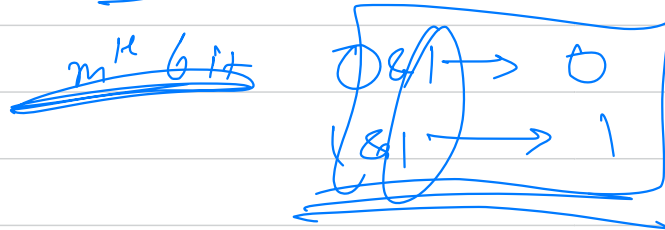
$n = n | (y')$



Q₂ Given a number n , clear the k^{th} bit

Set a bit \rightarrow make the value of bit as 1

Clear a bit \rightarrow make the value of bit as 0



$$\hookrightarrow 1 \ll k$$

$$n = 01001011 \quad k = 3$$

$$1 \ll 3 \Rightarrow \underline{\underline{1 \ll k}}$$

$$(1 \ll \cancel{k}) \rightarrow 00001000$$

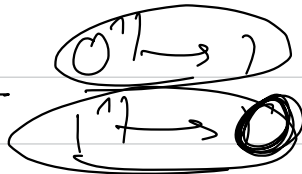
$\underbrace{\hspace{1.5cm}}_2 \quad \underbrace{\hspace{1.5cm}}_1$
 $\quad \quad \quad \downarrow \quad \quad \downarrow$
 $\quad \quad \quad 1 \quad \quad 1$

$$\underline{\underline{[\sim(1 \ll k) \ 1111011]}}$$

$$\underline{\underline{n \& \sim(1 \ll k)}}$$

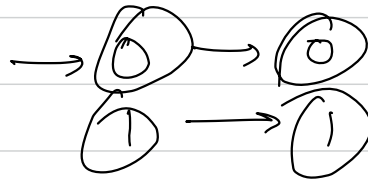
$$\begin{array}{r} 01001011 \\ 11110111 \\ \hline 01000011 \end{array}$$

K^m bits



xor

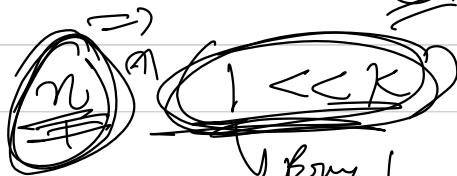
for rest
 m bits



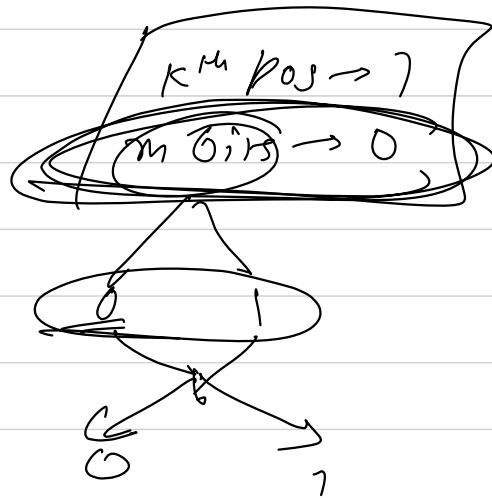
$0^m \mid \rightarrow 1$

$1^m \mid \rightarrow 0$

K^m bits \rightarrow 1



key 1 to K^m pos



Q → Given a number toggle the rightmost Set bit

↖ → zero

0001

1 < 3
1000 2^3

$n = 01001010$
 ans = 01001010

$n-1 = 01001010$

→ $n \& n-1$

0101
 0110

 0011

0 0 0 0
 0 0 0 1
 0 0 1 0 2^1
 0 0 1 1
 0 1 0 0 2^2
 0 1 0 1
 0 1 1 0
 0 1 1 1
 1 0 0 0 2^3
 1 0 0 1
 1 0 1 0
 1 0 1 1
 1 1 0 0

Q → Given a number n , clear all the bits except the rightmost set-bit

→ $n = 01001011$

$n \& -n$

Ans → 00000007

$n \& -n$

$-n$ → 2's complement of n

→ 10110100

→
10110100

Q-2 Multiply a given number n with a power of 2

$$(1 \leq k)$$

$$n \times 2^k$$

$$n \times (1 \leq k)$$

$$n \times 2^k$$

$$n \leq k$$

$$2^k$$

$$\rightarrow (n \times 2)^k$$

$$1 \leq k$$

$$100$$

$$4 \times 3$$

$$1100$$

$$+2$$

$$4 \times 5$$

$$20$$

$$10100$$

Q₂ Given a number n , Round it by $\frac{K^x}{1}$ power of 2

when $n \gg K$

Q₃ Round a number by 2 with '/' operator

→ $n \gg 1$

Qⁿ Check whether the number is a power of 2

In a power of two $(k+1)^{\text{th}}$ bit is set rest are clear

$(2^k - 1) \rightarrow (k+1)^{\text{th}}$ bit is clear, rest are set

$$\begin{array}{l} 2^3 \rightarrow \\ 2^3 - 1 = \end{array} \begin{array}{|c|c|c|c|} \hline 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 \\ \hline \end{array}$$

$$\begin{array}{l} 2^4 \rightarrow \\ 2^4 - 1 = \end{array} \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$\underbrace{n \& (n-1) == 0}_{\text{yes}}$$

Ex

$2^3 \rightarrow 1000$
$2^3 - 1 \rightarrow 0111$

$n-1$

0000

$(n \& n-1) \rightarrow \underline{\underline{0}}$

if $n \rightarrow \underline{\underline{2^k}}$

$2^4 \rightarrow 10000$

$2^4 - 1 \rightarrow 01111$

if $(n \& n-1 == 0)$

yes

else no

Q \Rightarrow Given a number n , count all the set bits in the binary of n .

total no. of bits \rightarrow
 $\log_2 n$

while $(n > 0)$ {
 count $+= (n \& 1) == 1$;
 $n = n >> 1$;
}

5 6 7 8 \rightarrow 1000
 $(\log_2 n + 1)$ 32 \rightarrow 1000000

Can we optimize??

$$n = \begin{array}{cccccccc} & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}$$

$$n = n \& n-1 \rightarrow \begin{array}{cccccccc} & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ & \text{---} & & & \text{---} & \text{---} & & \end{array}$$

$$n = n \& n-1 \rightarrow \begin{array}{cccccccc} & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ & & & & \text{---} & & & \end{array}$$

$$n = n \& n-1 \rightarrow \begin{array}{cccccccc} & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & \end{array}$$

$$n \& n-1 \rightarrow 00000000$$

$$n \rightarrow 0$$

8 byte 163

while (n > 0)

n = n & n - 1

count++

}
return count;

Brian
Kennighan's
Algo

4

31

O (no. of set bits)

unt → 1 byte
8 bits

2

-2^7 $2^7 - 1$

127
0 1 1 1 1 1 1 1

Q. You have an array $[1, 2, 3]$ $n \leq 10^5$

$n \rightarrow 2^n$

Calc the xor of all sub arrays

Ans $\rightarrow 0$

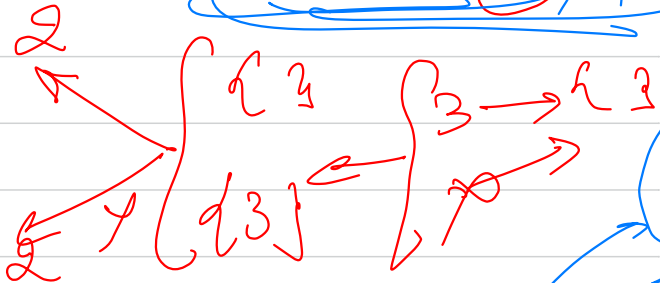
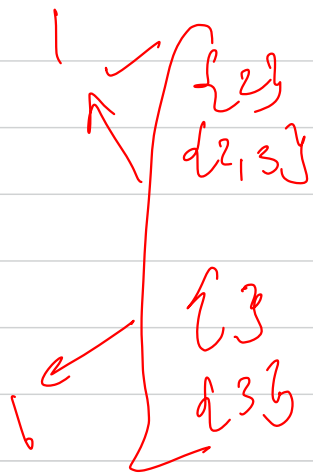
$([1] [1, 2] [1, 3] [1, 2, 3])$
 $([2] [2, 3] [1, 2, 3])$

$[1] [2] [3] [4] [5] [6]$
 $\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow$
 $2 \ 2 \ 2 \ 2 \ 2 \ 2$

2^5

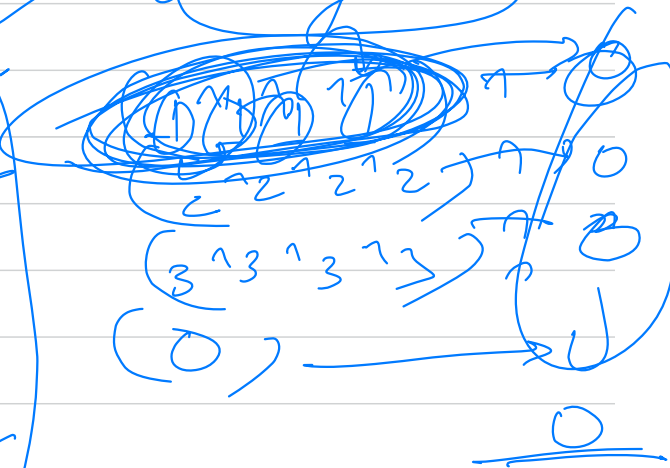
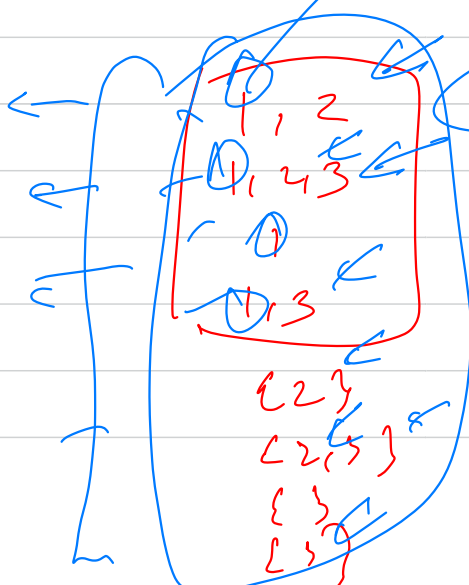
$\rightarrow 2^n$

1, 2, 3, 4



double

even no of sub



0