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

0 1 0 0 1 0 1 1 → A

0 0 0 1 0 1 0 1 → B

A & B → 0 0 0 0 0 0 0 1 } → 1

Bitwise OR (|)

	|	
0	0	$\rightarrow$ 0
1	0	$\rightarrow$ 1
0	1	$\rightarrow$ 1
1	1	$\rightarrow$ 1

2 2 2 2 2 2 2 2

0 1 0 0 1 0 1 1

0 0 0 1 0 1 0 1

A | B $\rightarrow$ 0 1 0 1 1 1 1 1

Bitwise Exclusive OR - XOR (^)

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

$$
\begin{array}{rl}
 & 0\;1\;0\;0\;1\;0\;1\;1 \\
 & 0\;0\;0\;1\;0\;1\;0\;1 \\
A \wedge B \rightarrow & \underline{0\;1\;0\;1\;1\;1\;1\;0}
\end{array}
$$

Bitwise Left Shift (<<) → It ~~fills~~ appends new 0 bits to the LSB side

→ You have an 8 bit number

5 → ~~0 0~~ 0 0 0 1 0 1 << 2

5 << 1 → 0 0 0 1 0 1 0 0

remove from MSB side

removed

x (0 1 0 0 1 0 1 1)

x << 2

→ 0 0 1 0 1 1 0 0

added

left side by appending zeroes &

removing MSB

Bitwise Right Shift (>>) → it prepends zeroes on the MSB side, & removes LSB.

x (0 1 0 0 1 0 $\underbrace{1\ 1}$) → removed

$x >> 2$

$\underbrace{0\ 0\ 0\ 1}\ \underbrace{0\ 0\ 1\ 0}$

added

Bitwise Complement (~) → It inverts the bits

x (0 1 0 0 1 0 1 1)

2nd bit

$\sim x$

→ 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.

0 <- 0 and 1

1 <- 1 and 1

=> ① Extract the value of K^{th} bit

② Check value

$K=2$

2nd bit

$K=2$

Decimal

X

0 1 0 0 1 0 1 1

00

$x >> K$

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

$1 \rightarrow$ (0 0 0 0 0 0 1)

if $((x' >> 2)\ \&\ 1)$

K^{th} bit was set

else unset

Q) Set the k^{th} bit

$k = 2^{nd}$

x' 0 1 0 0 1 0 1 1

Bitwise OR

y'

$1 \rightarrow 0\,0\,0\,0\,0\,0\,0\,1$

$y' (1 << 2) \rightarrow 0\,0\,0\,0\,0\,1\,0\,0$

$y' = 1 << k$

$x = x \mid (y')$

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

Set a bit → make the value of bit as 1

Clear a bit → make the value of bit as 0

k^{th} bit

$0 \& 0 \rightarrow 0$

$1 \& 0 \rightarrow 0$

m^{th} bit

$0 \& 1 \rightarrow 0$

$1 \& 1 \rightarrow 1$

$\hookrightarrow \quad 1 << K$

$n = 01001011 \qquad K = 3$

$1 << 3 \rightarrow 1 << K$

$(1 << K) \rightarrow \underbrace{0000}_{2}\ \underbrace{1}_{3}\ \underbrace{000}_{2}$

$[\sim(1 << K) \quad 1111\ 0\ 111]$

$n \ \& \ \sim(1 << K)$

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

k^{th} bit $\quad 0 \wedge 1 \to 1$
$\quad\quad\quad\quad 1 \wedge 1 \to 0$] XOR

for rest $\to$ $0 \to 0$
n bits $\quad\quad 1 \to 1$

$0 \wedge 0 \to 0$
$1 \wedge 0 \to 1$

$0 \wedge 1 \to 1$
$1 \wedge 1 \to 0$

k^{th} bit $\to 1$

$n \wedge (1 << k)$

bring 1 to k^{th} pos

k^{th} pos $\to 1$
n bits $\to 0$

0 1

0 1

Q => Given a number Toggle the rightmost Set bit

zero

0001

$1 << 3$

1000 2^3

$n = 0\,1\,0\,0\,1\,0\,1\,0$

ans = 0 1 0 0 1 0 1 0

$n-1 = 0\,1\,0\,0\,1\,0\,1\,0$

$\hookrightarrow (n \,\&\, n-1)$

0 1 0 1

0 1 1 0

0 0 1 1

0 → 0 0 0 0

0 0 0 1

2 → 0 0 1 0 2^1

0 0 1 1

4 → 0 1 0 0 2^2

0 1 0 1

6 → 0 1 1 0

0 1 1 1

8 → 1 0 0 0 2^3

9 → 1 0 0 1

10 → 1 0 1 0

11 → 1 0 1 1

1 1 0 0

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

↳ $n = 0\,1\,0\,0\,1\,0\,1\,1$

Ans → $0\,0\,0\,0\,0\,0\,0\,1$

$n \,\&\, n-1$

$n\,\&\,-n$ → 1

$-n$ → 2's complement of n

→ $1\,0\,1\,1\,0\,1\,0\,0$

→ 1

$1\,0\,1\,1\,0\,1\,0\,1$

Q-> Multiply a given number n with a k^{th} power of 2

$n \times 2^k$

4×3

$1 << k$

$n \times (1<<k)$

1100

$+2$

$n \times 2^k$

$\Rightarrow n << k$

4×5

2^k

$\rightarrow 1 \times 2^k$

100

20

101

10100

$1<<k$

Q-> Given a number n, divide it by 2^k power of 2 value $n >> k$

Q=> Divid a number by 2 with '>' operator

→ $n >> 1$

Qn Check whether the number is a power of **2**

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

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

$2^3 \rightarrow 1000$

$2^3 - 1 = 0111$

$2^4 \rightarrow 10000$

$2^4 - 1$ & 01111

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

Yes

n

$$\& \boxed{\begin{array}{l} 2^3 \rightarrow 1000 \\ 2^3-1 \rightarrow 0111 \end{array}}$$

$n-1$

$$\underline{0000}$$

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

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

$2^4 \rightarrow 10000$

$2^4-1 \rightarrow 01111$

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

<u>Yes</u>

else

<u>No</u>

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

total no. of bits

$\log_2 n$

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

5 6 7 8 → 1000

$\log_2 n + 1$

32 → 100000

Can we optimize??.

$n = 0\ 1\ 0\ 0\ 1\ 0\ 1\ 1$

$n = n \,\&\, n-1 \rightarrow 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0$

$n = n \,\&\, n-1 \rightarrow 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0$

$n = n \,\&\, n-1 \rightarrow 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0$

$n \,\&\, n-1 \rightarrow 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

$1 \rightarrow 0$

while (n > 0)

n = n & n-1

count++;

}

return count;

Brian Kernighan's Algo

8 byte

63

4

31

O(no. of set bits)

int → 1 byte

8 bits

2

-2^7 —— 2^7-1

127

0 1 1 1 1 1 1 1

$n \le 10^5$

Q: You have an array $\{1, 2, 3\}$

$n \to 2^n$

Calc the xor of all subsets

ans $\to 0$

$\{\ \}$ $\{1\}$ $\{2\}$ $\{3\}$ $\{1, 2\}$
$\{1, 3\}$ $\{2, 3\}$ $\{1, 2, 3\}$

2^5

1	2	3	4	5	6
2	2	2	2	2	2

$\to 2^n$

{1, 2, 3, 4}

doubly

even no of subset

1

{2}

{2, 3}

2

{ }

{3}

3 → { }

{3}

{3}

2

{1, 2}

{1, 2, 3}

{1}

{1, 3}

{2}

{2, 3}

{ }

{ }

0