

# \* Bit manipulation Operators

## ① AND

a	b	a & b
0	0	0
0	1	0
1	0	0
1	1	1

\* When you & 1 with any number, digits remains the same

$$\begin{array}{r}
 110010100 \\
 \& 111111111 \\
 \hline
 110010100
 \end{array}$$

## ② OR

a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1

## ③ XOR (^) (if and only if) exclusive OR

a	b	a ^ b
0	0	0
0	1	1
1	0	1
1	1	0

### observations

$$a \wedge 1 = \bar{a}$$

$$a \wedge 0 = a$$

$$a \wedge a = 0$$

# \* Bit manipulation Operators

## ① AND

a	b	$a \& b$
0	0	0
0	1	0
1	0	0
1	1	1

\* When you  $\&$  with any number, digits remains the same

$$\begin{array}{r}
 110010100 \\
 \& 111111111 \\
 \hline
 110010100
 \end{array}$$

## ② OR

a	b	$a \text{ OR } b$
0	0	0
0	1	1
1	0	1
1	1	1

## ③ XOR ( $\wedge$ ) (if and only if) exclusive OR

a	b	$a \wedge b$
0	0	0
0	1	1
1	0	1
1	1	0

## observations

$$a \wedge 1 = \bar{a}$$

$$a \wedge 0 = a$$

$$a \wedge a = 0$$

## \* Bit manipulation

## Operators

## ① AND

a	b	a & b
0	0	0
0	1	0
1	0	0
1	1	1

\* When you & 1 with any number, digits remains the same

$$\begin{array}{r}
 110010100 \\
 \& 111111111 \\
 \hline
 110010100
 \end{array}$$

## ② OR

a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1

③ XOR (^) (if and only if)  
exclusive OR

a	b	a ^ b
0	0	0
0	1	1
1	0	1
1	1	0

observations

$$a \wedge 1 = \bar{a}$$

$$a \wedge 0 = a$$

$$a \wedge a = 0$$

④ Complement ( $\sim$ )

$$a = 10110$$

$$\bar{a} = 01001$$

\* Number Systems

① Decimal  $\rightarrow 0, 1, 2, \dots, 9$  Base 10

$$(357)_{10}, (10)_{10}$$

② Binary  $\rightarrow 0 \text{ \& } 1$  Base 2

$$(10)_{10} = (1010)_2$$

$$(7)_{10} = (111)_2$$

③ Octal  $\rightarrow 0, 1, 2, 3, \dots, 7$  Base 8

Decimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Octal: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20

$$(9)_{10} = (11)_8$$

④ Hexadecimal: -  
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F  
Base: 16

$$(10)_{10} = (A)_{16}$$

$$(12)_{10} = (C)_{16}$$

Conversation

① Decimal to base b

Q Convert  $(17)_{10}$  to base 2

Keep dividing by base, take remainder, write in opposite

$$\begin{array}{r}
 2 \overline{) 17} \\
 \underline{2 \phantom{0} 8} \phantom{0} 1 \\
 2 \phantom{0} \underline{4} \phantom{0} 0 \\
 2 \phantom{0} \underline{2} \phantom{0} 0 \\
 \underline{1} \phantom{0} 0
 \end{array}$$

$$(10001)_2 = (17)_{10}$$

$$(17)_{10} = (8)_{\cancel{8}}$$

$$\begin{array}{r}
 8 \overline{) 17} \\
 \underline{2}
 \end{array}$$

② Convert any base b to decimal

$$(10001)_2 = ( \quad )_{10} ?$$

Steps :- multiply &amp; add the power of base with digits

$$= 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 16 + 1,$$

$$= 17$$



$$9 \quad (21)_8 = (\quad)_{10}$$

$$= 2 \times 8^1 + 1 \times 8^0$$

$$= 2 \times 8 + 1$$

$$= (17)_{10}$$

\* Continuing with operators

③ left shift operator ( $\ll$ )

$$(10)_{10} = (1010)_2 \quad 10 \ll 1$$

step:  $1010 \ll 1 = 10100$

$$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$= 16 + 4$$

$$= 20$$

$$\therefore a \ll 1 = 2a$$

General point:

$$a \ll b = a \times 2^b$$

④ Right shift  $\gg$

$$0011001 \gg 1 \Rightarrow 001100$$

$$(00011234)_{16} = (11234)_{10} \Rightarrow (1100)$$

(1 ignored)  $\rightarrow$  same for all number systems

General:  
point

$$a \gg b = \frac{a}{2^b}$$

Questions:-

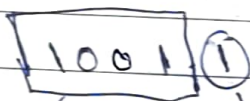
- ① Given a no.  $n$  find if it is odd or even.  
Point:- Every no is calculated in binary form internally

$$12 + 7 \Rightarrow \begin{array}{r} 1100 \\ + 0111 \\ \hline 10011 \end{array}$$

$(19)_{10} = (10011)_2$

$$16 + 2 + 1$$

Note:-



leaving this  
every other is  
a power of 2

This will always be even.

Hence if  $2^0$  place  $= 1 \Rightarrow$  odd  
otherwise  $\Rightarrow$  even

# code

```
static boolean isOdd(int n) {  
    return (n & 1) == 1;  
}
```

②  $arr = [2, 3, 4, 1, 2, 1, 3, \textcircled{6}, 4]$

Ans

Find unique element?

ans: XOR all the nos.

# Code

```
static int ans(int[] arr) {
    int unique = 0;

    for (int n : arr) {
        unique ^= n;
    }

    return unique;
}
```

9 Find  $i^{th}$  bit of a no.

Ans

8 7 6 5 4 3 2 1  
1 0 1 1 0 1 1 0  
          ?

1 0 1 1 0 1 1 0  
0 0 0 1 0 0 0 0  
1 0 0 0 1 0 0 0 Ans

This is called  
a mask

$n \Rightarrow$  mask with  $n-1$  zero  $\rightarrow 1 \ll (n-1)$

$1 \ll 4 \Rightarrow 10000$

Ans:  $n \& (1 \ll (n-1))$



Q Set the  $i$ th bit

turn it to 1

$i$ th

$$\begin{array}{r} 1010110 \\ \text{OR } 0001000 \\ \hline 1011110 \end{array} \rightarrow \text{mask}$$

Q Reset  $i$ th bit

1  $\rightarrow$  0

0  $\rightarrow$  0

$$\begin{array}{r} 1010110 \\ \& 1101111 \\ \hline 1000110 \end{array} \rightarrow \text{How to get this mask?}$$

Ans

$$\text{mask} : 1 \ll (n-1)$$

Q

Find the position of the right most set bit

$$\text{Ex:- } \begin{array}{r} 101101100 \\ \underline{\quad\quad\quad} \quad \underline{\quad\quad} \\ a \quad \quad \quad b \end{array} \quad \text{Ans} = 4$$

$$N = a \mid b$$

$$a = 101101$$

$$b = 00$$

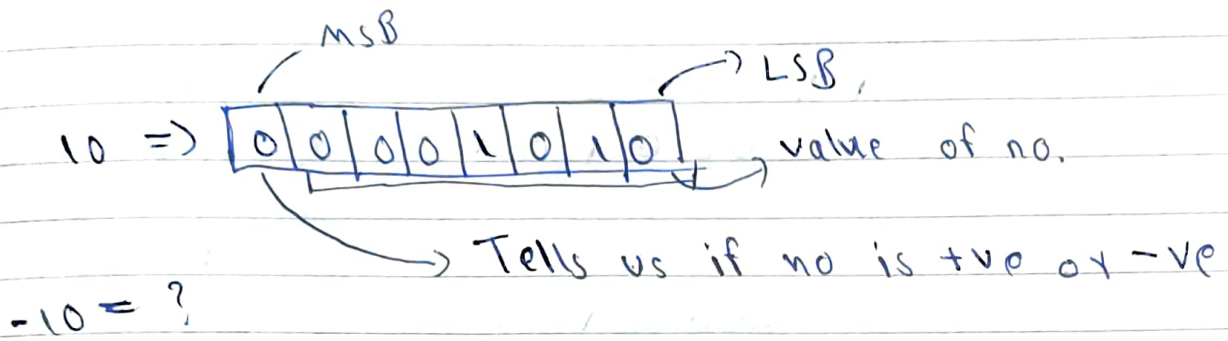
$$\sim N = \sim a \mid b$$

How?

$$\text{Ans} = N \& (\sim N)$$

\* Negative of a number in Binary form:

$$1 \text{ byte} = 8 \text{ bits}$$

1  $\rightarrow$  -ve0  $\rightarrow$  +ve

steps :-

- ① Complement of no.
- ② +1 to it

$\Rightarrow$  aka  $2^s$  complement method

$$(10)_{10} = (00001010)_2$$

①  $11110101$

②  $11110101$

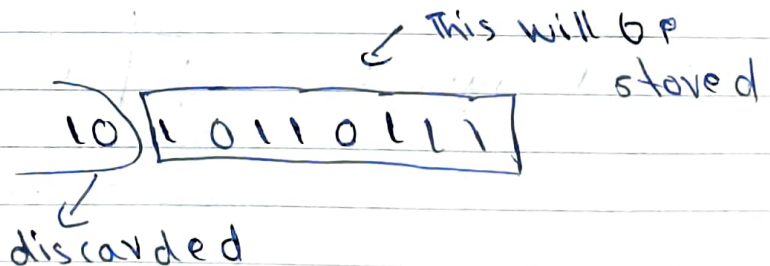
$$+ \quad 1$$


---


$$(11110110)_2$$

$$\Rightarrow (-10)_{10}$$

Why tho?



\* When we subtract any number by 0 then we will get -ve of that number

10 in binary  $00001010$

$$\begin{array}{r}
 10000000 \\
 - 00001010 \\
 \hline
 \end{array}$$

What is this

$$\begin{array}{r}
 1000 = 111 + 1 \\
 8 = 7 + 1
 \end{array}$$

$$\begin{array}{r}
 10000 = 1111 + 1 \\
 16 = 15 + 1
 \end{array}$$

$$100000000 = 11111111 + 1$$

$$\begin{array}{r}
 1111 \\
 + 0001 \\
 \hline
 10000
 \end{array}$$

Now:  $11111111 + 1 = 00001010$

$$\Rightarrow 11111111 - 00001010 + 1$$

complement

Step 1

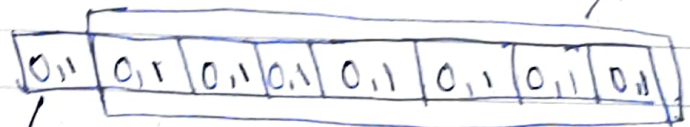
Step 2

$$\begin{array}{r}
 11111111 \\
 - 00001010 \\
 \hline
 11110101
 \end{array}$$

complement

\* Range of numbers:

① 1 byte:



sign of number

$$\begin{aligned}
 \text{Total} &= 2 \times 2 \times 2 \times 2 \dots \text{8 times} \\
 &= 2^8 = 256
 \end{aligned}$$

actual no is stored in bits =  $n-1$

↳ total bits

In 1 byte: 7 bits

Total I can make from 7 bits  
 $= 2^7 = 128$

- 128 to 127

→ 0 is here as well

\* -0 is 0

proof

00000000

11111111

complement

Adding 1 111111

11111111

+ 1

100000000

~~discarded~~ discarded

9-bit

discarded

\* Range formula: for  $n$  bits

$$[-2^{n-1} \text{ to } 2^{n-1} - 1]$$

9 arr = [2, 2, 3, 2, 7, 7, 8, 7, 8, 8]

[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

1 0

1 1

1 0

1 1 1

1 1 1

1 0 0 0

1 1 1

1 0 0 0

$$\boxed{3 \mid 3 \mid 7 \mid 4} \% 3$$

$$\boxed{0 \ 0 \ 1 \ 1} = 3$$

Q Find the  $n^{\text{th}}$  magic no.

$$1 = \begin{matrix} s^3 & s^2 & s^1 \end{matrix} \begin{matrix} 0 & 0 & 1 \end{matrix} \longrightarrow \text{magic no } 5$$

$$2 = \begin{matrix} s^3 & s^2 & s^1 \end{matrix} \begin{matrix} 0 & 1 & 0 \end{matrix} \longrightarrow 25$$

$$3 = \begin{matrix} s^3 & s^2 & s^1 \end{matrix} \begin{matrix} 0 & 1 & 1 \end{matrix} \longrightarrow 30 \ (5+25)$$

$$4 = \begin{matrix} s^3 & s^2 & s^1 \end{matrix} \begin{matrix} 1 & 0 & 0 \end{matrix} \longrightarrow 125$$

$$5 = \begin{matrix} s^3 & s^2 & s^1 \end{matrix} \begin{matrix} 1 & 0 & 1 \end{matrix} \longrightarrow 130$$

Q  $n = 6$   $110$

loop  $\left[ \begin{array}{l} n \& 1 \Rightarrow \text{This will give me last digit in binary} \\ n >> 1 \end{array} \right.$

$$0 \times s^3 + 1 \times s^2 + 1 \times s^1$$

Q Find no of digit in base 6

$$(6)_{10} \Rightarrow 1$$

$$(6)_{10} \Rightarrow (110)_2 = 3 \text{ (By keeping counter)}$$



Formula:- No of digits in base  $b$  of no.  $x$

$$= \text{int}(\log_b x) + 1$$

$$\log_b a = \frac{\log_x a}{\log_x b}$$

$$\log_b a = x$$

$$a = b^x$$

$$\log_2 6 = x$$

$$6 = 2^x$$

$$\log_2 10 = 3.32$$

$$10 = 2^{3.32}$$

$\rightarrow \text{int} + 1$   
= no of digits

$$\begin{array}{ccccccc} & & & & 1 & & 1 \\ & & & 1 & & 2 & & 1 \\ & & 1 & & 3 & & 3 & & 1 \\ & 1 & & 4 & & 6 & & 4 & & 1 \end{array}$$

\* Pascal's Triangle

$\therefore$  Find the sum of  $n^{\text{th}}$  row

Sum of each row =

$${}^nC_0 + {}^nC_1 + {}^nC_2 + \dots + {}^nC_n = 2^n$$

For  $n^{\text{th}}$  row, sum =  $2^{n-1}$

Ans:

$$1 < (n-1) = 1 * 2^{n-1}$$

Ans

Q You are given a number. Find out if its power of 2 or not.

100000, 100010

$$1000000 = 1111111 + 1$$

if its power of 2

$$\begin{array}{r} 10000000 \\ \& 01111111 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 10010 \\ \& 01111 \\ \hline 00010 \end{array}$$

Ans: If  $n \& (n-1) = 0$  // It is power of 2  
Q Calculate  $a^b$

$$3^6 \Rightarrow 3 * 3 * 3 * 3 * 3 * 3$$

$$3^6 = 3^{110} = 3^{2*4} = 3^2 * 3^4$$

ans = 1	n = 110	ans = 1
	$n \& 1 \Rightarrow 0$	ans * base
base = 3	$n > 1 \Rightarrow 11 \& 1 = 1$	
base = 9	$n = 11 > 1 \Rightarrow 1 \& 1 = 1$	
base = base	$n = 1 > 1 = 0$	
* base		

$$O(\log(b))$$

$$3^{110} \Rightarrow 3 * 1^4 * 3 * 1^2 * 3 * 1^0$$

Imp to understand this question

Q Given a number  $n$ , find the no of set bits in it.

ans  $n = 9$   
 $n = 1001$  Ans = 2

$$n \& (-n) = 0001$$

$$n - [n \& (-n)] = 1000 \Rightarrow (1)$$

$$\begin{array}{r} n = 1001 \\ \& 1000 \\ \hline 1000 \end{array} \quad (1)$$

$$8 \& 7 \Rightarrow 1000$$

$$\begin{array}{r} \& 111 \\ \hline 0 \end{array} \quad (2)$$

No of set bits = no of iterations

Q Find XOR of nos from 0 to  $a$

a XOR from 0 to  $a$

0	0	6	7
1	$0^1 = 1$	7	0
2	$0^1 \wedge 2 = 3$	8	8
3	0	9	10
4	4		
5	1		
6	7		

~~order~~  
~~as~~  
~~d~~

If  $a \% 4 = 0$   
 $a \% 4 = 1$   
 $a \% 4 = 2$   
 $a \% 4 = 3$

$0 \rightarrow a$   
 $a$   
 $1$   
 $a+1$   
 $0$

Q XOR of all nos between a & b

$a = 3$  &  $b = 9$

$3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 8 \wedge 9$

$0 \wedge 1 \wedge 2 \wedge 3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 8 \wedge 9$

These are the extras

This is  $0 \rightarrow (a-1)$

Ans  $f(b) \wedge f(a-1)$   
 $f(n) \rightarrow \text{XOR of } 0 \rightarrow x$

		1	2	3	4	5
up = row	1	1	1	1	1	1
down = 2n-1	2	1	2	2	2	1
right = N-col	3	1	2	3	2	1
left = col	4	1	2	2	2	1
	5	1	1	1	1	1