PW SKILLS

Bit Manipulation

Checklist

PW SKILLS

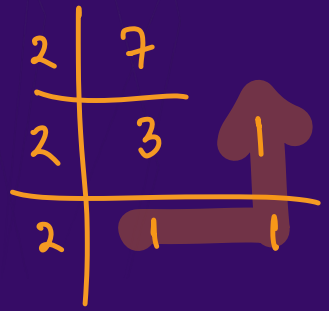
1. Binary and Decimal conversions
2. Bitwise operators - &, |, ^, >>, << (and overflow), ~(NOT), >>>
3. Tricks and Questions. Power of Operators

Bit Manipulation is one of the most important topics for interviews

Decimal to Binary Conversion

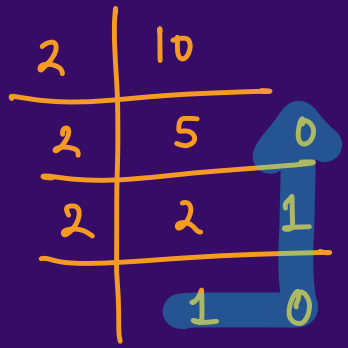
SKILLS

$n = (7)_{10}$



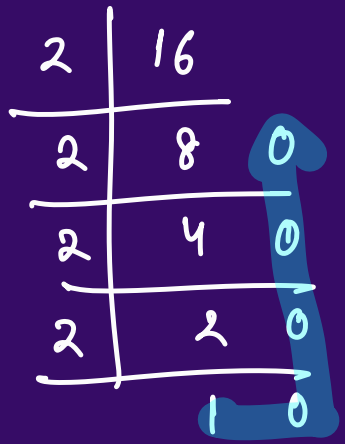
111 is 7

$n = (10)_{10}$



1010 is 10

$n = (16)_{10}$



16 is 10000

Binary to Decimal Conversion

PW SKILLS

$$(1\ 0\ 1\ 1\ 1\ 0)_2$$

$$0\times2^0 + 1\times2^1 + 1\times2^2 + 1\times2^3 + 0\times2^4 + 1\times2^5$$

$$= 2 + 4 + 8 + 32$$

$$= 46$$

$$(1\ 0\ 0\ 0)_2$$

$$2^3 = 8$$

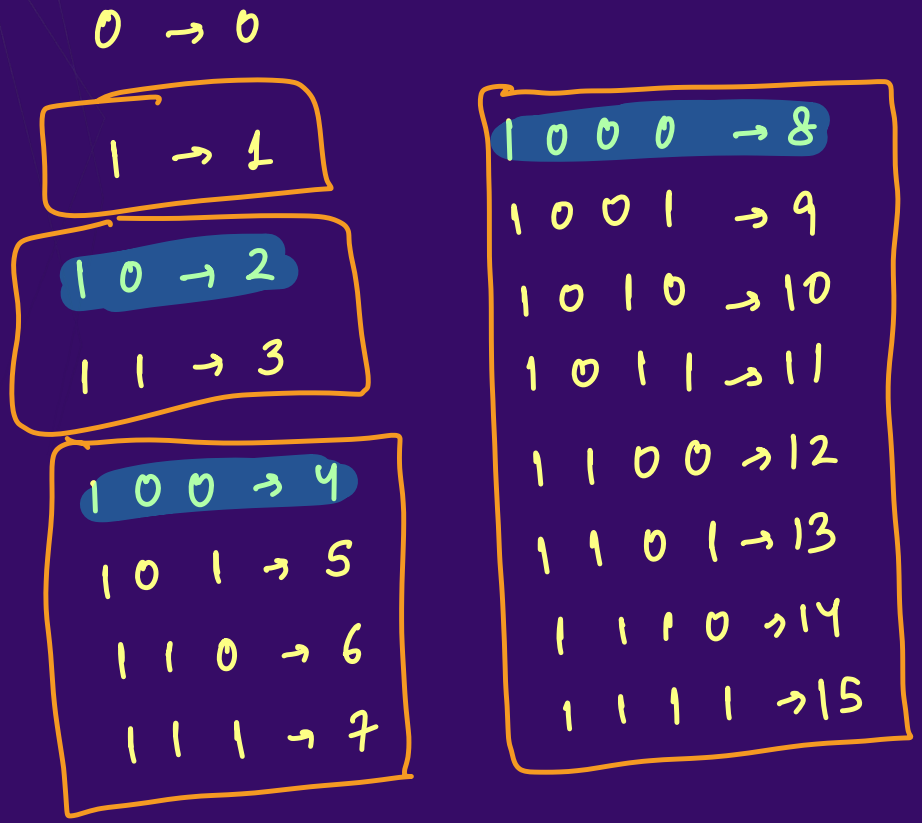
$$(1\ 1\ 1\ 1)_2$$

$$2^0 + 2^1 + 2^2 + 2^3$$

$$= 1 + 2 + 4 + 8$$

$$= 15$$

Binary to Decimal Conversion



$64 = 2^6$

1000000

~~Binary~~ to ~~Decimal~~ Conversion

PW SKILLS

Decimal to Binary

$n = 14$

$14 = 8 + 4 + 2$

$$\begin{array}{r} 1000 \\ 0100 \\ + \quad 0010 \\ \hline 1110 \end{array}$$

Binary Addition → Right to left

$$\begin{array}{rl} 1111 & \\ 1011 & \rightarrow 11 \\ + \quad 1101 & \rightarrow 13 \\ \hline 11000 & \end{array}$$

24

$16 + 8 = 24$

↓

11000

$1\underbrace{0000\ldots0}_{\text{all zeros}} = 2^K$

$1\,1\,1\,1\,1\,1\,1\,1 \rightarrow 2^t - 1$

Bitwise AND, Bitwise OR

PW SKILLS

&& , ||

↓ Logical and

Logical or

& , |

↓ Bitwise and

Bitwise or

Bitwise Or

0 | 0 = 0

0 | 1 = 1

1 | 0 = 1

1 | 1 = 1

Bitwise And

0 & 0 = 0

0 & 1 = 0

1 & 0 = 0

1 & 1 = 1

Bitwise AND, Bitwise OR

cout (5 | 9);

```
  0 1 0 1
| 1 0 0 1
---------
  1 1 0 1
```

convert to decimal

13

cout (5 & 11)

```
  0 1 0 1
& 1 0 1 1
---------
  0 0 0 1
```

Bitwise AND, Bitwise OR

$x|0 = x$
$x\&0 = 0$

PW SKILLS

$11 \rightarrow 1011$

$a|p = 1$

$a\ 0\ c\ d$
OR $p\ 0\ r\ s$
$1\ 0\ 1\ 1$

$1|0 = 1$
$0|1 = 1$
$1|1 = 1$

1011
1011
$11|11 = 11$

1011
0011
$11|3 = 11$

1011
0000
$11|0 = 11$

1001
0010
$9|2 = 11$

Bitwise AND, Bitwise OR

SKILLS

Data Types :

byte → -2^7 to 2^7-1 → -128 to 127

short → -2^{15} to $2^{15}-1$ → -32768 to 32767

int → -2^{31} to $2^{31}-1$ →

long → -2^{63} to $2^{63}-1$

Binary Number me bits hoti hai

1 bit → 2 numbers ko store kar sakta hai → 0 or 1

1 byte = 8 bits

0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1

$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^8 = 256$ options

Bitwise AND, Bitwise OR

int → can store 2^{32} numbers → -2^{31} to $2^{31}-1$

byte → 2^8 numbers → -128 to 127

byte x = 17

0	0	0	1	0	0	0	1

17 → 10001

int x = 17

~~Bitwise AND, Bitwise OR~~

byte x = 127

leftmost bit is for sign

0 → +ve

1 → -ve

Bitwise XOR

PW SKILLS

$\wedge$ (XOR)

$1 \wedge 1 = 0$

$1 \wedge 0 = 1$

$0 \wedge 1 = 1$

$0 \wedge 0 = 0$

Even number of ones $\rightarrow$ 0

Odd number of ones $\rightarrow$ 1

---

a | b | c | d

b | a | d | c

Associative hote hai bitwise operators

1's Complement, 2's Complement

flip all bits

1 0 1 1 0 0 1

↓ 1's Complement

0 1 0 0 1 1 0

---

x, ~x = -x-1

---

6 ka 1's Complement

~ → tilde

` → backtick

2's complement of x is -x

2's comp(x) = 1's Comp(x) + 1

4 bits

nibble, byte, short, int, long

How -ve numbers are stored.

PW SKILLS

nibble x = 7;

'nibble' ki range → -8 to 7

2^4 numbers

— — — —

↑ -8 to 7

nibble, byte, short, int, long

How -ve numbers are stored.

0 000	0
0 001	1
0 010	2
0 011	3
0 100	4
0 101	5
0 110	6
0 111	7

+ve ✓

1 000	-0
1 001	-1
1 010	-2
1 011	-3
1 100	-4
1 101	-5
1 110	-6
1 111	-7

-ve ✗

Ese store nahi hote

nibble, byte, short, int, long

How -ve numbers are stored.

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

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

$-5 = \sim 5 + 1$

$$
\begin{array}{r}
0\ 1\ 0\ 1 \\
\hline
1\ 0\ 1\ 0 \\
+\ 0\ 0\ 0\ 1 \\
\hline
1\ 0\ 1\ 1
\end{array}
$$

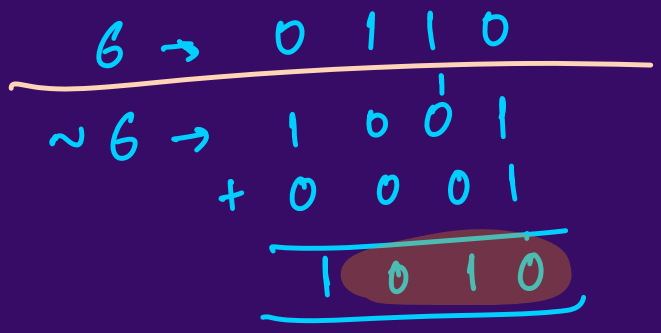
nibble, byte, short, int, long

How -ve numbers are stored.

$-x = \sim x + 1$

nibble a = -6

$-6 = \sim 6 + 1$



-ve decimal numbers conversion to binary

Binary to Decimal (Negetive)

Behavior of Operators

Behavior of Operators

PW SKILLS

$a \rightarrow 0, 1$

$a \wedge 0 = a$

$a \wedge 1 =$ flipped a

$a | 0 = a$

$a | 1 = 1$

to turn on a bit

$a \& 1 = a$

$a \& 0 = 0$

to turn off a bit

$a * b \rightarrow$ T.C. $= O(1)$

$*$ ⇒ $|, \&, \wedge, >>, <<, \sim$

Left Shift, Right Shift

PW SKILLS

$a = 1\ 0\ 1\ 1\ 0\ 1\ 1 \quad (91)_{10}$

$a >> 2 \Rightarrow 0\ 0\ 1\ 0\ 1\ 1\ 0 \quad (22)_{10}$

↓

no. of bits shifted/removed from right

$a >> 3 \quad 1\ 0\ 1\ 1 \quad (11)_{10}$

Left Shift, Right Shift

PW SKILLS

$a = 00\,10\,11\,011$ $(91)_{10}$

$a << 2 \Rightarrow 10\,11\,01\,100$ $(364)_{10}$

byte $x = 0\,1\,1\,0\,0\,1\,1\,0$ $(102)_{10}$

$x << 3 \Rightarrow 0\,0\,1\,1\,0\,0\,0\,0$ $(48)_{10}$

$0\,1\,1\,0\,0\,1\,1\,0\,0\,0\,0$ $(816)_{10}$

512 256 32 16

Left Shift, Right Shift

1

Given 'n'. Find 2^n. $\Rightarrow O(1)$

2^0 1

2^1 10

2^2 100

2^3 1000

2^4 10000

2^5 100000

Binary

2^n = 1 00.... 0 (n zeros)

$2^n = 1 << n$

T.C. = $O(1)$

Ques:

Q : Swap 2 numbers (XOR)

1.
```
temp = a
a = b
b = temp
```

2.
```
a = a + b
b = a - b
a = a - b
```

3.
```
a = a ^ b
b = a ^ b
a = a ^ b
```

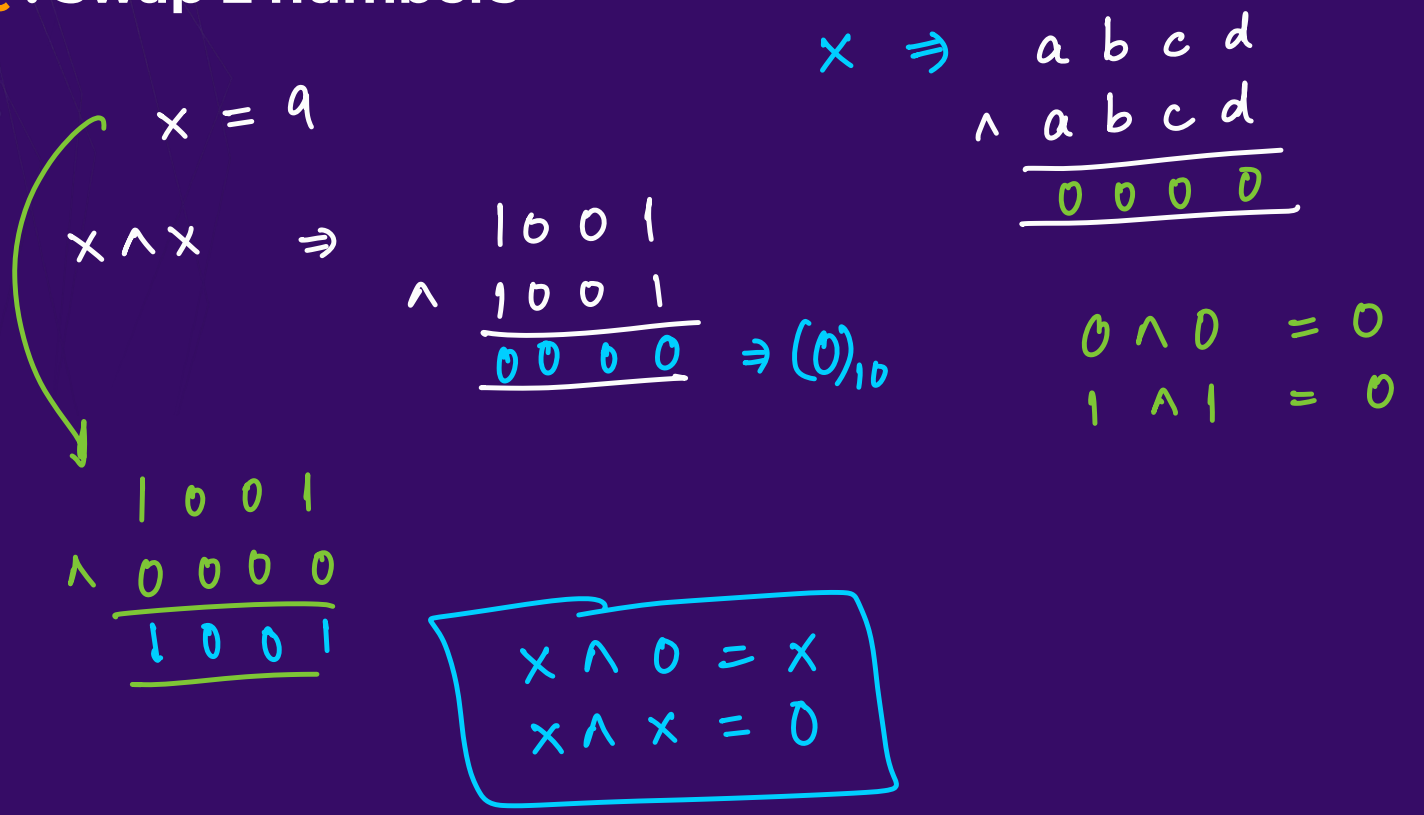
$x \wedge x = 0$

$x \wedge 0 = x$

XOR is associative

Ques:

Q : Swap 2 numbers



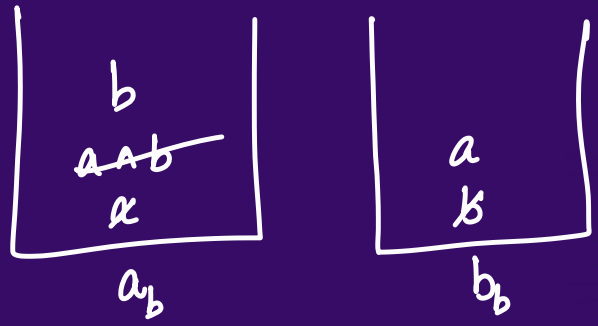
Ques:

Q : Swap 2 numbers

$a = a \wedge b$

$b = a \wedge b$

$$\begin{aligned} b &= (a \wedge b) \wedge b \\ &= a \wedge (b \wedge b) \\ &= a \wedge 0 \\ &= a \end{aligned}$$



$a = a \wedge b$

$$\begin{aligned} a &= (a \wedge b) \wedge a \\ &= b \wedge (a \wedge a) \\ &= b \wedge 0 \\ &= b \end{aligned}$$

Ques:

PW SKILLS

Q: Single Number

arr = [576, 2, 3, 576, 99, 3, 99]

M-I: Brute Force → T.C. = $O(n^2)$, A.S = $O(1)$

M-II: Sorting T.C. = $O(n \log n)$, A.S. = $O(\log n)$

M-III: Hashmap T.C. = $O(n)$, A.S. = $O(n)$

M-IV: Using XOR operator. T.C. = $O(n)$, A.S. = $O(1)$

[Leetcode 136]

Ques:

PW SKILLS

Q: Single Number

arr = { 4, 1, 2, 3, 1, 4, 2 }

↓ sort

= { 1, 1, 2, 2, 3, 4, 4 }

i

---

arr = { 4, 1, 2, 3, 1, 3, 2 }

↓ sort

= { 1, 1, 2, 2, 3, 3, 4 }

i

Ques:

PW SKILLS

Q : Single Number

arr = { 4, 1, 2, 3, 1, 4, 2 }

4 ^ 1 ^ 2 ^ 3 ^ 1 ^ 4 ^ 2

4 ^ 4 ^ 1 ^ 1 ^ 2 ^ 2 ^ 3

0 ^ 0 ^ 0 ^ 3

0 ^ 3

3

Ques:

1 hai ya nahi

'Bitmasking'

SKILLS

Q : Check if kth bit is set or not.

```
          10 9 8 7 6 5 4 3 2 1 0
a    =    1 0 1 1 1 0 0 0 0 1 1
&
mask =    0 0 0 0 0 0 0 1 0 0 0   →   1<<3
          ---------------------
          0 0 0 0 0 0 0 0 0 0 0
```

K = 3

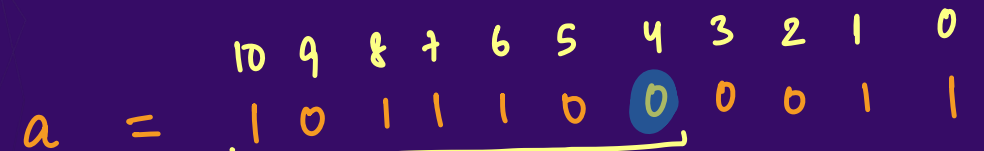
if (a & mask == 0) → bit was off / not set (false)

if (a & mask != 0) → set hai (true)

```
return (a & (1<<k) != 0);
```

Ques:

Q : Check if kth bit is set or not.

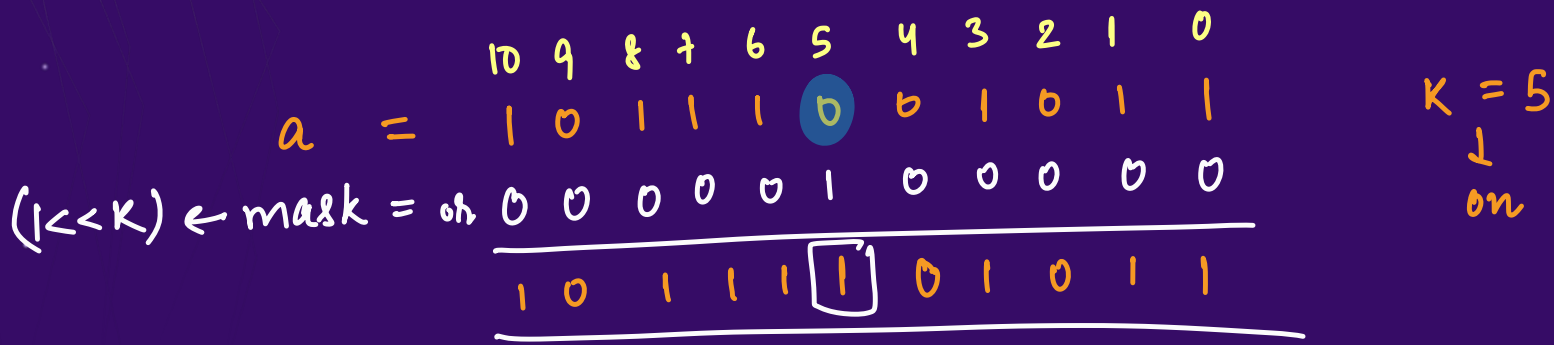



```
b = a >> k ;
```


```
return (b%2 == 1)
```

Ques:

Q : Turn on / Set the kth bit




```
a = a | mask
```

```
a = a | (1<<K)
```

Ques:

Q : Turn off / Clear the kth bit

K = 7

a & 1 = a

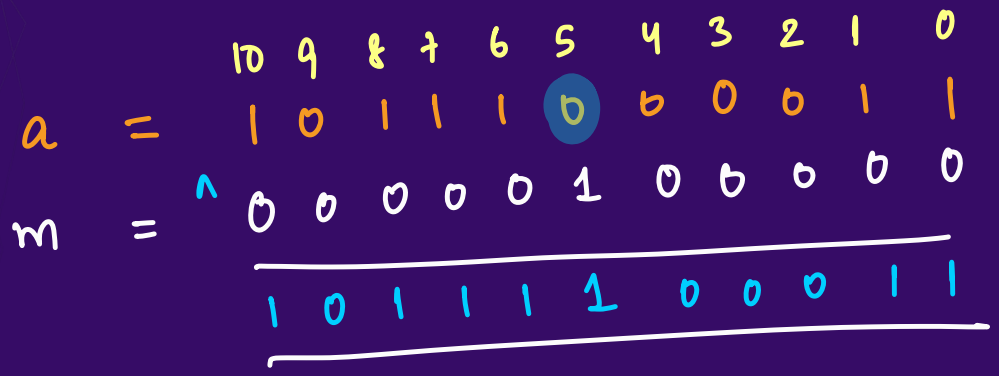
a & 0 = 0

mask = ~(1<<K)

a = a & mask

Ques:

Q : Toggle the kth bit (XOR)



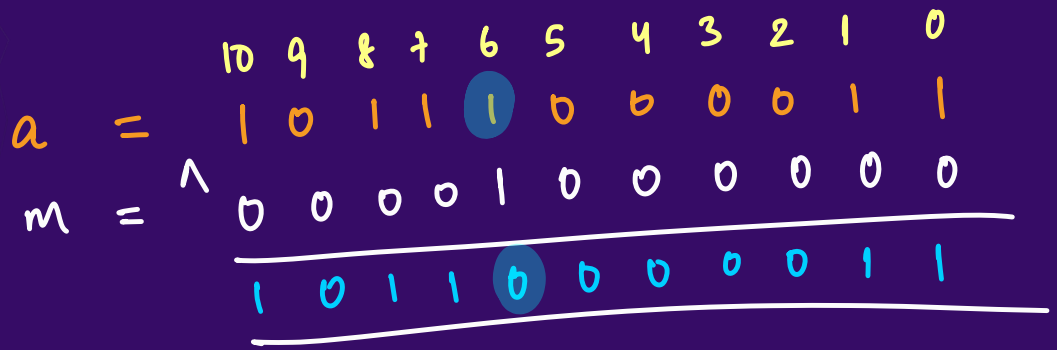
K = 5
toggle

$m = 1 << K$

$a = a \wedge m$

Ques:

Q : Toggle the kth bit



K = 6

Ques: turn off the [a = 1 0 0 0 0 0 0 0 0 0 0]

PW SKILLS

Q : Rightmost set bit

Method-1 (Find 'k')

a = 1 0 1 1 1 0 0 1 0 0 0

k = 0 1 2 3

b = 1 0 1 1 1 0 0 0

```
b = a
k = 0
while (b % 2 == 0) {
    b >> 1;
    k++;
}
```

T.C. = $O(\log_2 n)$ / $O(31)$

Ques:

Q : Rightmost set bit → turn off the

`n & (n-1)`

`n = n & n-1` ⇒ O(1) T.C

PW SKILLS

Method - 2 :

n = 32

100000

n = 31

011111

n & n-1 ⇒ 000000

n = 43

101001

n = 42

101000

n & n-1 ⇒ 101000

n = 42

101000

n-1 = 41

100111

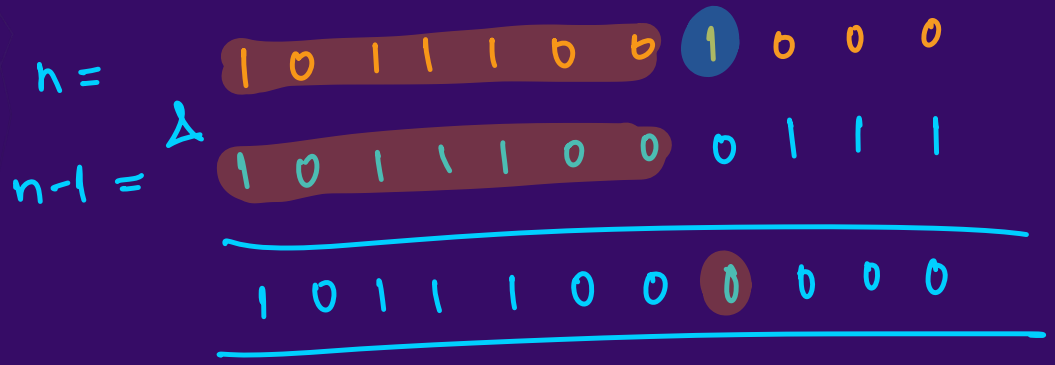
&

100000

Ques:

Turn off

Q : Rightmost set bit



Ques:

PW SKILLS

Q : Power of Two

Given n. Check if it is a power of 2 or not

n = 64 true

n = 60 false.

Method-1 Recursion

↓

$n \rightarrow n/2 \rightarrow n/4 \ldots 1$

T.C. = $O(\log_2 n)$

[Leetcode 231]

Ques:

Hint n, n-1

PW SKILLS

Q : Power of Two

Method - 2 '&' operator

T.C. = O(1)

16 = 10000
15 = 01111
16 & 15 = 0 true

64 = 1000000
63 = 0111111
64 & 63 = 0

n=15

1 1 1 1
1 1 1 0
& 1 1 1 0 != 0000

false.

[Leetcode 231]

Ques:

Q : Number of 1 Bits

(Best)

Method-1 : Turning off the rightmost set bit again & again.

cnt

0

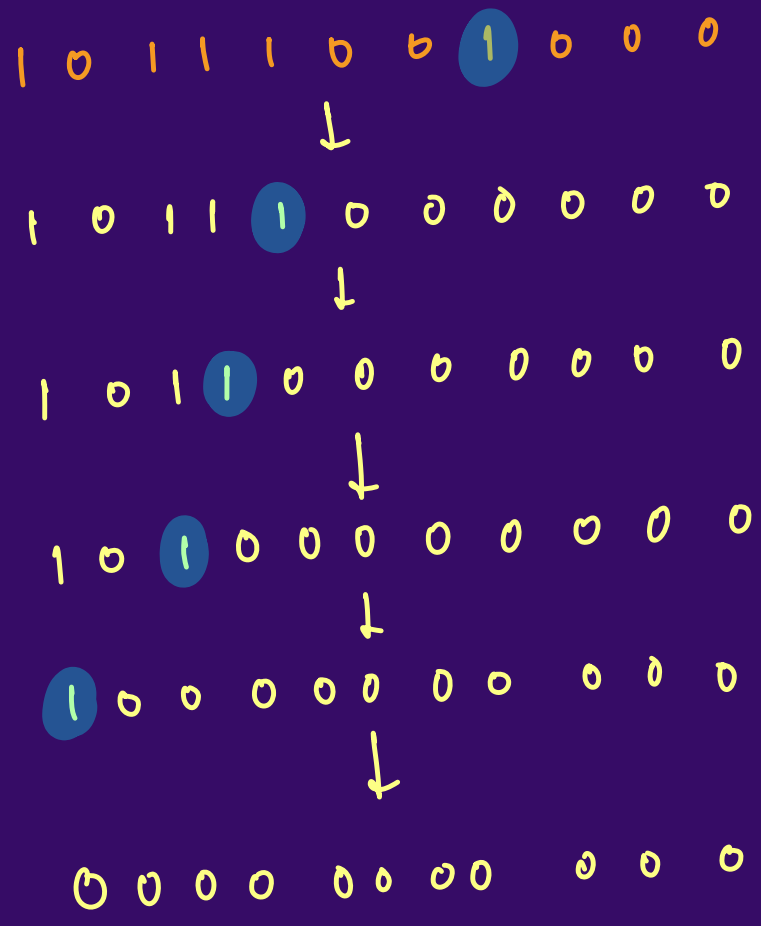
1

2

3

4

5



T.C. = O(cnt)

[Leetcode 191]

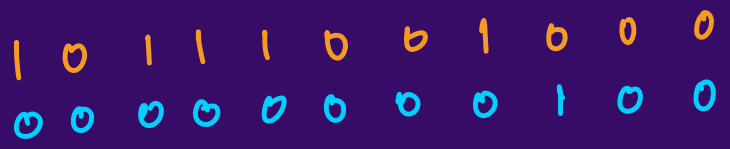
Ques:

PW SKILLS

Q : Number of 1 Bits

Method-2 : T.C. = O(32) Brute Force

count
0



you will create 32 masks

& you will check if the i^{th} bit is set or not

$1 << i \rightarrow i = 0$ to 31

[Leetcode 191]

Ques:

$a \wedge b = 0$ if a&b both are 0
a&b both are 1

PW SKILLS

Q : Minimum Bit Flips to convert Number

start goal [no. of ones in start ^ goal]

13 18

Start = 14
goal = 15

```
  1 1 1 0
^ 1 1 1 1
-----------
  0 0 0 1
```

ans = 1 flip

```
  0 1 1 0 1
^ 1 0 0 1 0
-----------
  1 1 1 1 1
```

st
go

ans = 5 flips

[Leetcode 2220]

Ques:

PW SKILLS

Q : Power of Four → 1, 4, 16, 64, 256 . .

M-1 : Brute Force (Recursion)
Keep dividing by 4. → T.C. = O(log n)

$1 << K \Rightarrow 2^K$

$n \to \frac{n}{4} \to \frac{n}{16} \cdots 1 \to \log_4 n$

[Leetcode 342]

Ques:

PW SKILLS

Q : Power of Four

- If any number is power of four, then 100% it is a power of two as well but vice-versa is not true.

- All odd powers of 2 [2^n where n is odd] are not powers of 4.

2, 8, 32, 128 . . . these are not 4^K

$4^K = (2^2)^K = (2^K)^2$ which is a perfect square

[Leetcode 342]

Ques: $(1+x)^n = 1 + {}^nC_1 x^1 + {}^nC_2 x^2 + {}^nC_3 x^3 + \ldots$

PW SKILLS

Q : Power of Four

- Any power of four leaves a remainder '1' when divided by 3.

For Ex →

$1 \% 3 = 1$

$4 \% 3 = 1$

$16 \% 3 = 1$

$64 \% 3 = 1$

$256 \% 3 = 1$

$4^n = (1+3)^n = 1 + {}^nC_1 \cdot 3^1 + {}^nC_2 \cdot 3^2 + {}^nC_3 \cdot 3^3 \ldots + {}^nC_n \cdot 3^n$

$= 1 + 3p$

[Leetcode 342]

Ques:

PW SKILLS

Q : Single Number III

In the given array, 2 numbers appear once, rest all appear twice.

Method-1: Using Hashmap

T.C. = O(n)

A.S. = O(n)

Method-2: Sorting

T.C. = O(n log n)

A.S. = O(log n)

[Leetcode 260]

Ques:

PW SKILLS

Q : Single Number III

Method-3 : Bit Manipulation (XOR & Rightmost Set Bit)

T.C. = O(n)
A.S. = O(1)

arr = { 1, 5, 7, 1, 8, 5, 9, 7, 2, 2 }

1 ^ 1 ^ 5 ^ 5 ^ 7 ^ 7 ^ 2 ^ 2 ^ 8 ^ 9

```
  1000
^ 1001
------
  0001
------
```

xor = 1

- 2 unique elements are different, so atleast 1 bit will be different

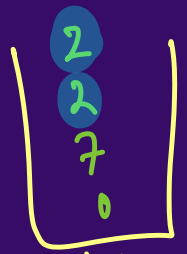
[Leetcode 260]

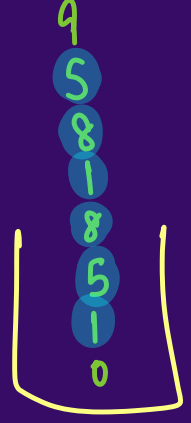
Ques:

Q : Single Number III

arr = { 1, 5, 8, 1, 8, 5, 9, 7, 2, 2 }

$x = 7 \wedge 9$





$x \wedge (x-1)$

0 1 1 1
$\wedge$ 1 0 0 1
1 1 1 0
1 1 0 1
1 1 0 0
mask 0 0 1 0

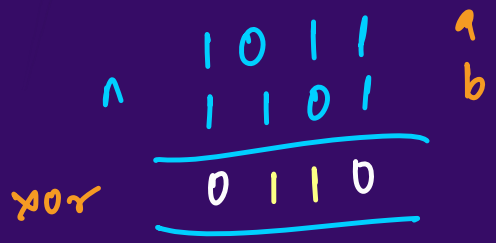
[Leetcode 260]

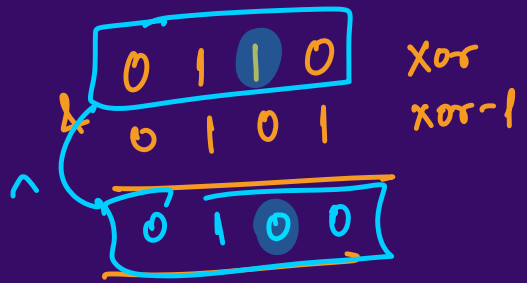
Ques:

Q : Single Number III

a & b

- XOR of two number is a number & in this number all ones represent diff bits in a & b





0010

I want this : 0010

mask

[Leetcode 260]

Ques:

PW SKILLS

Q : Subsets

n elements in a set can form 2^n subsets.

{a, b, c}

[a, b] ⇒ [], [a], [b], [a, b]

[a, b, c] ⇒ [[],
[a], [b], [c]
[a, b], [a, c], [b, c]
[a, b, c]]

[Leetcode 78]

Ques:

Q : Subsets

bit - index
ko
array index
se
map karana hai

$n = 3$ $\quad 2^n = 1 << n$

	a	b	c			
0	0	0	0	—	—	—
1	0	0	1	—	—	c
2	0	1	0	—	b	—
3	0	1	1	—	b	c
4	1	0	0	a	—	—
5	1	0	1	a	—	c
6	1	1	0	a	b	—
7	1	1	1	a	b	c

T.C. = $O(n * 2^n)$

S.C. = $O(n * 2^n)$

[Leetcode 78]

Ques:

Q : Reverse Bits

: Brute Force Solution

Convert integer to Binary Array (Size 32)

$T.C. = O(32)$

$A.S. = O(32)$

n = 23

10111

11101

29

5 bits only

2	23	
2	11	1
2	5	1
2	2	1
	1	0

11101

[Leetcode 190]

Ques:

PW SKILLS

Q : Reverse Bits

1) Bit Manipulation Method

for 'int', the bit index goes from 0 to 31

$i = 0, j = 31$

T.C. = O(32)

A.S. = O(1)

[Leetcode 190]

Ques:

PW SKILLS

Q : XOR of numbers in a given Range $[a, b]$

$[2, 9] \Rightarrow 2 \wedge 3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 8 \wedge 9$

Method-1 : Brute Force → Loop

T.C. = $O(n)$

- XOR of 1 to n

  observation & dry run question. →

Ques:

Q : XOR of numbers in a given Range

n =		
1	=	1
2	=	3
3	=	0
4	=	4
5	=	1
6	=	7
7	=	0
8	=	8
9	=	1
10	=	11
11	=	0
12	=	12
13	=	1
14	=	15
15	=	0
16	=	16

n%4 == 1	1
n%4 == 2	n+1
n%3 == 3	0
n%4 == 0	n

Ques:

Q : XOR of numbers in a given Range

n = 77 ⇒ 1

77 % 4 = 1

n = 87 ⇒ 0

87 % 4 = 3

n = 62 ⇒ 63

62 % 4 = 2

O(1)

```
xor (int n){
    if(n%4==1) return 1;
    if (n%4==2) return n+1;
    if (n%3==3) return 0;
    if (n%4==0) return n;
}
```

Ques:

PW SKILLS

Q : XOR of numbers in a given Range

Find the XOR from 3 to 9

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

$= (1 \wedge 2) \wedge (1 \wedge 2 \wedge 3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 8 \wedge 9)$

$xor(a,b) = xor(1, a-1) \wedge xor(1, b)$

Ques:

PW SKILLS

Q : XOR queries of a Subarray

arr = {1, 3, 4, 8} (indices 0 1 2 3)

n-size

queries = {(0,1), (1,2), (0,3), (3,3)} (m size)

ans = { 2, 7, 14, 8 }

M-I : Brute Force

T.C. = O(m*n) (TLE Error)

[Leetcode 1310]

Ques:

PW SKILLS

Q : XOR queries of a Subarray

M-2 : Using prefix sum concept.

arr = { 1, 3, 4, 8 }

pre = { 1, 2, 6, 14 }

(1, 3) xor batao

T.C. = O(m+n)

(0,3) → 14

(0, 0) → 1

1 ^ 3 ^ 4 ^ 8

[Leetcode 1310]

Homework:

PW SKILLS

Q : XOR Operation in an Array {O(1) Time}

Hint : observation

[Leetcode 1486]