Todays Content:

a) Number System Basics

b) Binary to Decimal & viceversa

c) Adding 2 Binary numbers

d) Bit Wise operations

    i) Basic Properties

    ii) Basic Problems

Decimal Number System

→ Each Digit : [0 1 2 3 4 5 6 7 8 9]

→ Each power : [10]

$\underline{10^3}\ \underline{10^2}\ \underline{10^1}\ \underline{10^0}$

3 4 2 = 300 + 40 + 2 = $3*10^2 + 4*10^1 + 2*10^0$

2 5 6 3 = 2000 + 500 + 60 + 3 = $2*10^3 + 5*10^2 + 6*10^1 + 3*10^0$

2 4 5 = 200 + 40 + 2 = $2*10^2 + 4*10^1 + 5*10^0$

Binary Number System

→ Each Digit : [0 1]

→ Each power : [2]

$\underline{2^4}\ \underline{2^3}\ \underline{2^2}\ \underline{2^1}\ \underline{2^0}$

1 0 1 1 0 : $2^4*1 + 2^3*0 + 2^2*1 + 2^1*1 + 2^0*0 = 22$

1 0 1 0 : $2^3*1 + 2^2*0 + 2^1*1 + 2^0*0 = 10$

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

1 0 1 0 0 : $2^4*1 + 2^3*0 + 2^2*1 + 2^1*0 + 2^0*0 = 20$

1 2 0 : Not a Binary Number. digit [0 1]

Decimal to Binary

rem

- 2 | 37 - 1
- 2 | 18 - 0
- 2 | 9 - 1
- 2 | 4 - 0
- 2 | 2 - 0
- 2 | 1 - 1
- 0

rem

- 2 | 25 - 1
- 2 | 12 - 0
- 2 | 6 - 0
- 2 | 3 - 1
- 2 | 1 - 1
- 0

rem

- 2 | 19 - 1
- 2 | 9 - 1
- 2 | 4 - 0
- 2 | 2 - 0
- 2 | 1 - 1
- 0

$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

37 : 0 1 0 0 1 0 1

: $2^5 + 2^2 + 2^0$

: 32 + 4 + 1 = 37

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

25 : 0 1 1 0 0 1

: $2^4 + 2^3 + 2^0$

: 16 + 8 + 1 = 25

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

19 : 0 1 0 0 1 1

: $2^4 + 2^1 + 2^0$

= 16 + 2 + 1 = 19.

Add 2 decimal numbers: d = s % 10   c = s / 10   : 10 → decimal numbers

Ex 1:

```
      1    1
      7    8    9
 0    1    4    2
 -----------------
      9   13   11
      9    3    1
```

Ex 2:

```
 c:     1    0    1
        7    8    3    9
 1      3    9    4    8
 ------------------------
   11%10 17%10 8%10 17%10
   1    1    7    8    7
```

Add 2 Binary Numbers d = s % 2   c = s / 2   : 2 → Binary numbers

```
c = 0   0    1    1    0
        1    0    1    1    0
        0    0    1    1    1
        -------------------------
s =     1%2  1%2  3%2  2%2  1%2
d = 0   1    1    1    0    1
```

```
c = 1   1    0    1    0
        1    1    0    1    1
        1    1    0    1    0
        -------------------------
s =     3%2  2%2  1%2  2%2  1%2
d = 1   1    0    1    0    1
```

Why Binary?

In electronics current passes
if voltage → TODO
- >= certain limit : 1
- < certain limit : 0

int n = 25;

In system data stored in Binary → **Decimal to Binary**

| n = 11001 |

→ **Binary to Decimal**

print(n) : 25

These internal conversions are internally done by your system.
We don't have to worry.

Bitwise operations : { AND, OR, XOR, Invert, leftshift, right shift }

& | ^ ~ << >>

A	B	A & B	A | B	A ^ B	~ A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

: If diff: 1
If same: 0

Same Same puppy Shame

// a = 29, b = 19

	2^4	2^3	2^2	2^1	2^0		Decimal
a :	1	1	1	0	1		
b :	1	0	0	1	1		
print(a & b) :	1	0	0	0	1	→	17
print(a | b) :	1	1	1	1	1	→	31
print(a ^ b) :	0	1	1	1	0	→	14

// a = 13, b = 10

	2^3	2^2	2^1	2^0		Decimal
a :	1	1	0	1		
b :	1	0	1	0		
print(a & b):	1	0	0	0	→	8
print(a | b):	1	1	1	1	→	15
print(a ^ b):	0	1	1	1	→	7

Bit Wise Properties

// 0^{th} bit : Bit with power 2^0

1. $A = 10$:

	2^3	2^2	2^1	2^0	
A	1	0	1	0	dec:
1 :	0	0	0	1	
A & 1 :	0	0	0	0	0

2. $A = 14$:

	2^3	2^2	2^1	2^0	
A	1	1	1	0	dec:
1 :	0	0	0	1	
A & 1 :	0	0	0	0	0

3. $A = 11$:

	2^3	2^2	2^1	2^0	
A	1	0	1	1	dec:
1 :	0	0	0	1	
A & 1 :	0	0	0	1	$2^0 * 1 = 1$

4. $A = 13$:

	2^3	2^2	2^1	2^0	
A	1	1	0	1	dec
1 :	0	0	0	1	
A & 1 :	0	0	0	1	$2^0 * 1 = 1$

Obs: A & 1
- 0 : 0^{th} Bit in A = 0 : For even number 0^{th} bit is 0
- 1 : 0^{th} Bit in A = 1 : For odd number 0^{th} bit is 1

Ex:

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
A :	1	0	1	1	0	0	1	0

$= 2^7 * 1 + 2^5 * 1 + 2^4 * 1 \ldots 2^1 * 1$

$= 128 + 32 + 16 + 2 \ldots$

= Sum of even = even

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
A :	1	0	1	1	0	0	1	1

$= 2^7 * 1 + 2^5 * 1 + 2^4 * 1 \ldots 2^1 * 1 + 2^0 * 1$

= even + 1 = odd

Few More Interesting Properties

1. A & 1 → 0: A is even   1: A is odd

2. A & 0 → A : 1010 : 0
   0 : 0000
   A & 0 : 0000

3. A & A → A : 1010 : A
   A : 1010
   A & A : 1010

A | 1 → TODO

4. A | 0 → A : 1010 : A
   0 : 0000
   A | 0 : 1010

5. A | A → A : 1010 : A
   A : 1010
   A | A : 1010

A ^ 1 → TODO

6. A ^ 0 → A : 1010 : A
   0 : 0000
   A ^ 0 : 1010

^ : Same : 0 Diff : 1

7. A ^ A → A : 1010 : 0
   A : 1010
   A ^ A : 0000

Commutative Property : Order does not matter

True/False

1. $A \& B = B \& A$ True

2. $A \mid B = B \mid A$ True

3. $A \wedge B = B \wedge A$ True

1. $A \& B \& C = B \& C \& A = C \& B \& A = C \& A \& B = B \& A \& C = A \& C \& B$
2. $A \mid B \mid C = B \mid C \mid A = C \mid B \mid A = C \mid A \mid B = B \mid A \mid C = A \mid C \mid B$
3. $A \wedge B \wedge C = B \wedge C \wedge A = C \wedge B \wedge A = C \wedge A \wedge B = B \wedge A \wedge C = A \wedge C \wedge B$

Eg: 1. $a \wedge b \wedge a \wedge d \wedge b : a \wedge a \wedge b \wedge b \wedge d = d$ $a \wedge 0 = a$ $a \wedge a = 0$

2. $d \wedge e \wedge f \wedge a \wedge a \wedge d \wedge f : a \wedge a \wedge d \wedge d \wedge f \wedge f \wedge e = e$

3. $1 \wedge 3 \wedge 5 \wedge 3 \wedge 2 \wedge 1 \wedge 5 = 1 \wedge 1 \wedge 3 \wedge 3 \wedge 5 \wedge 5 \wedge 2 = 2$

$1 : 0\ 0\ 1$
$3 : 0\ 1\ 1$
$1 \wedge 3 : 0\ 1\ 0$
$5 : 1\ 0\ 1$
$1 \wedge 3 \wedge 5 : 1\ 1\ 1$
$3 : 0\ 1\ 1$
$1 \wedge 3 \wedge 5 \wedge 3 : 1\ 0\ 0$
$2 : 0\ 1\ 0$
$1 \wedge 3 \wedge 5 \wedge 3 \wedge 2 : 1\ 1\ 0$
$1 : 0\ 0\ 1$
$1 \wedge 3 \wedge 5 \wedge 3 \wedge 2 \wedge 1 : 1\ 1\ 1$

$1 \wedge 3 \wedge 5 \wedge 3 \wedge 2 \wedge 1 : 1\ 1\ 1$
$5 : 1\ 0\ 1$
$1 \wedge 3 \wedge 5 \wedge 3 \wedge 2 \wedge 1 \wedge 5 : 0\ 1\ 0 = 2$

Q) Given ar[N], where every element repeats twice except for 1 element, which occurs once, find that unique element.

Constraints:

$1 <= N <= 10^5$

$1 <= ar[i] <= 10^9$

	0	1	2	3	4		ans
Ex1: ar[5] =	6 ^	9 ^	6 ^	10 ^	9	=	10

	0	1	2	3	4	5	6		ans
Ex2: ar[7] =	2 ^	3 ^	5 ^	6 ^	3 ^	6 ^	2	=	5

Idea: Take xor of all ele.

```
int unique(int ar[], int n){          TC: O(N)  SC: O(1)
    int ele = 0 // xor with 0 won't effect your value
    i = 0; i < n; i++){
        ele = ele ^ ar[i]
    }
    return ele
}
```

	0	1	2	3	4		ans
Ex1: ar[5] =	6 ^	9 ^	6 ^	10 ^	9	=	10

ele	i^{th} : ele = ele ^ ar[i]	ele
0	0: ele = 0 ^ 6	6
6	1: ele = 6 ^ 9	15
15	2: ele = 15 ^ 6	9
9	3: ele = 9 ^ 10	3
3	4: ele = 3 ^ 9	10

leftshift <<

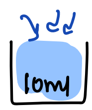
Say a is 8 bit number.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		
a = 25 :	0	0	0	1	1	0	0	1	= 16+8+1 = 25	a = 25
a<<1 :	0	0	1	1	0	0	1	0	= 32+16+2 = 50	a<<1 : $25 * 2^1$
a<<2 :	0	1	1	0	0	1	0	0	= 64+32+4 = 100	a<<2 : $25 * 2^2$
a<<3 :	1	1	0	0	1	0	0	0	= 128+64+8 = 200	a<<3 : $25 * 2^3$
a<<4 :	1	0	0	1	0	0	0	0	= 128+16 = 144	* a<<4 = $25*2^4$ = 400

dis dis dis dis

add add add add

loosing Inf:? Because overflow.



: When we store, more than it's limit, overflow occurs.

Obs: $a << n = a * 2^n$: Valid, if no overflow

a=1 $1 << n = 1*2^n = 2^n$

a=5 $5 << n = 5*2^n$

Doubt: When overflow occurs? Saturdays Class

Right shift >>

Say a is 8 bit number.

		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0			
$a = 25$:		0	0	0	1	1	0	0	1		$= 16+8+1 = 25$	$a = 25$
$a>>1$:	add	0	0	0	0	1	1	0	0	dis	$= 8+4 = 12$	$a>>1 = 25/2^1$
$a>>2$:	add	0	0	0	0	0	1	1	0	dis	$= 4+2 = 6$	$a>>2 = 25/2^2$
$a>>3$:	add	0	0	0	0	0	0	1	1	dis	$= 2+1 = 3$	$a>>3 = 25/2^3$
$a>>4$:	add	0	0	0	0	0	0	0	1	dis	$= 1$	$a>>4 = 25/2^4$
$a>>5$:	add	0	0	0	0	0	0	0	0	dis	$= 0$	$a>>5 = 25/2^5$
$a>>6$:	add	0	0	0	0	0	0	0	0	dis	$= 0$	$a>>6 = 25/2^6 = 0$

Obs: $a>>n = a/2^n$