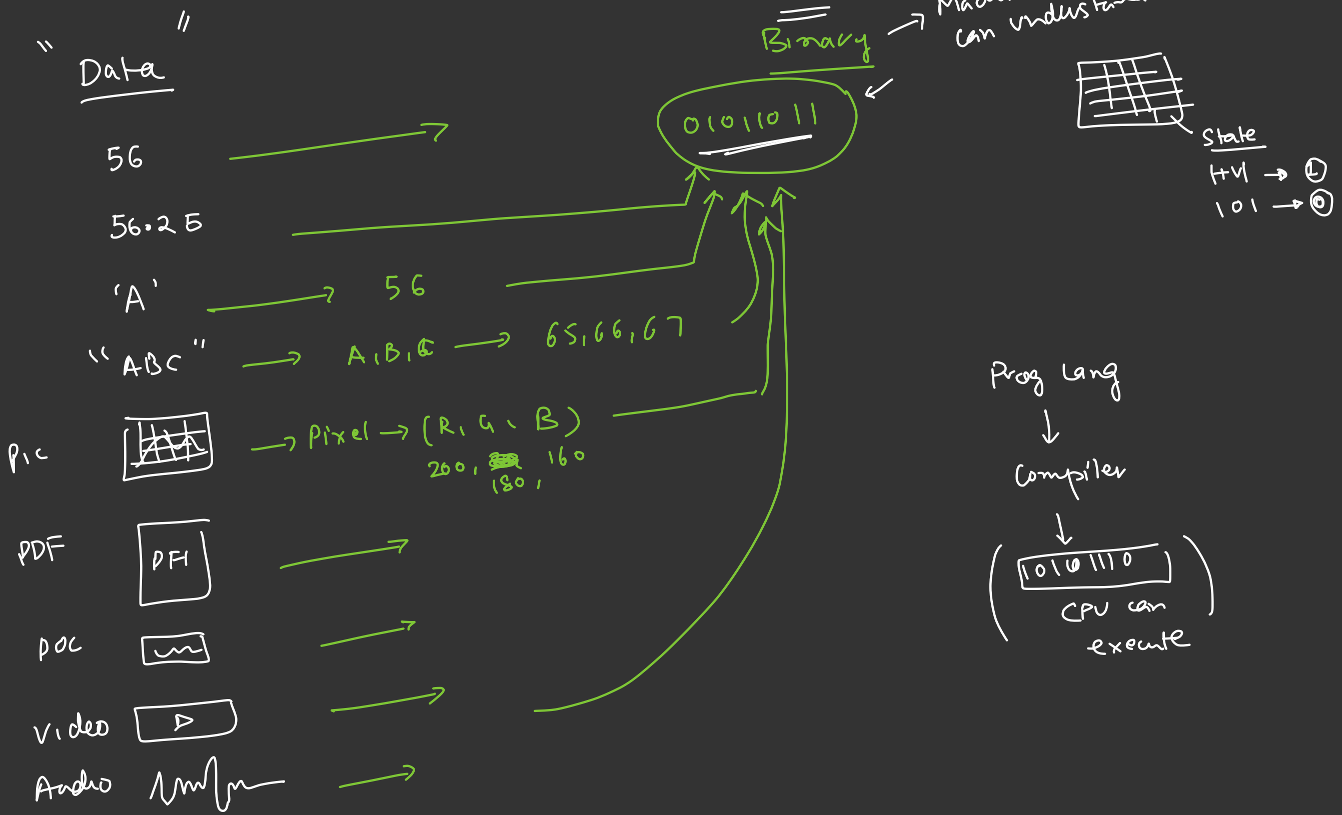
Bit manipulation



Computer Science

Numbers ⟶ Binary System

"Decimal" No System

↓

10 unique digits

Base 10

$\overbrace{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}$

Powers of 10

$\overset{10^2}{1}, \overset{10^1}{4}, \overset{10^0}{6} \rightarrow 10^2 \cdot 1 + 10^1 \cdot 4 + 10^0 \cdot 6$

$= 100 + 40 + 6$

$= 100 + 40 + 6$

$= \boxed{146}$ ← Magnitude

Example

Base-5

0, 1, 2, 3, 4

⟷

$2\cancel{4}9$ → is it valid base 5 No?

↳ invalid

$$\overset{5^2}{1}\ \overset{5^1}{2}\ \overset{5^0}{4}$$ → valid no

$$5^2 \cdot 1 + 5^1 \cdot 2 + 5^0 \cdot 4$$

$$= 25 + 5 \cdot 2 + 1 \cdot 4$$

$$= 25 + 10 + 4 = \boxed{39}$$

[Base10]

Binary No System

Base-2

Powers of 2

0, 1

120 → invalid binary No

⇒ 1011 → ✓ valid

$$\overset{2^3}{1}\ \overset{2^2}{0}\ \overset{2^1}{1}\ \overset{2^0}{1}$$

$$= 2^3 \cdot 1 + 2^2 \cdot 0 + 2^1\ 1 + 2^0 \cdot 1$$

$$= 8 + 0 + 2 + 1$$

$$= \boxed{11}$$

Examples

Binary → Decimal No

101 ⇒ $\boxed{5}$

1100 ⇒ $\boxed{12}$

1111 ⇒ $\boxed{15}$

011 ⇒ $\boxed{3}$

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1
		1	0	1
	1	1	0	0
	1	1	1	1
		0	1	1
	1	1	0	1

Algorithm:

int no = $\boxed{1101}$;

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

1 → 2 → 4 → 8

p → 2p → 4p → 8p

//13 ✓

no = 1 1 0 1 with 2^3 2^2 2^1 2^0 above

power = 1 ~~8~~

ans = 0 +

```
while ( no > 0 ) {          no==0 → stop
    last_digit = no%10 ;
    ans = ans + last_digit * power
    no = no/10 ; ←
    power = 2* power ←
}
print(ans);
```

ans = 0 + 1*1 + 0*2 + 1*4 + 1*8 = 0 + 1 + 0 + 4 + 8

= $\boxed{13}$

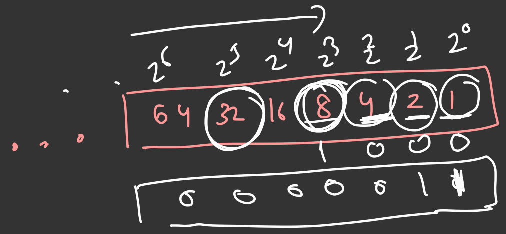
pow = 1 → 2 → 4 → 8 → 16

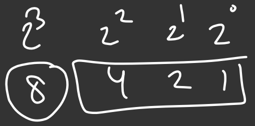
no = 1101
↓
110
↓
11
↓
1
↓
0 → stop

Decimal to Binary Conversion

Decimal			
1	→	1	✓
2	→	10	✗
3	→	11	✓
4	→	100	✗
5	→	101	✓
6	→	110	✗
7	→	111	✓
8	→	1000	✗
9	→	1001	✓
10	→	1010	✗
11	→	1011	✓
12	→	1100	✗
13	→	1101	✓

Hack





$\boxed{40} = 101000$

$$\begin{array}{cccccc} 1 & 0 & 1 & 0 & 0 & 0 \\ 32 & 16 & 8 & 4 & 2 & 1 \end{array}$$

$\underline{56} = $

$$\begin{array}{ccccccc} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array}$$

||

$32 + 16 + 8 = 56$

$\boxed{78}$

$\uparrow$

64 +

256	128	64	32	16	8	4	2	1
0	0	1	0	0	1	1	1	0

$64 + 8 + 4 + 2$

$= 78$ ✓

Intutive way

~~78~~ ~~14~~

$78 - 64$

$= \boxed{14} - 8$

$= \boxed{6} - 4$

$= 2 - 2 = 0$

Algoritmm → Repeated div by 2.

2	78	
2	39,	0
2	19,	1
2	9.	1

→ Remainders.

Rem

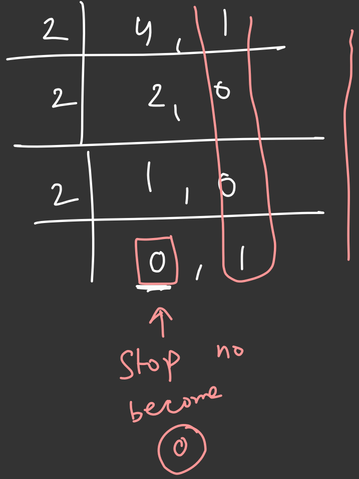
$78 = 2 \times \underline{39} + \underline{0}$

$39 = 2 \times \underline{19} + \underline{1}$

64	32	16	8	4	2	1
1	0	0	1	1	1	0

→ $64 + 8 + 4 + 2$

$+ 0$



2	4, 1
2	2, 0
2	1, 0
	0, 1

Stop no become 0

= 78

1 = 2 × 0 + 1

Reverse

0, 1, 1, 1, 0, 0, 1

output

Algorithm

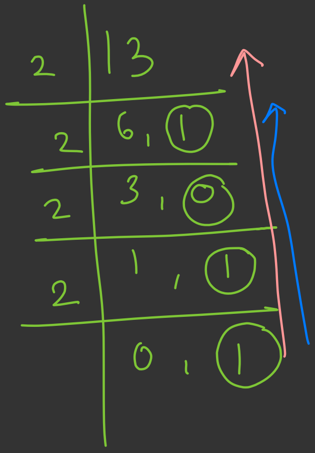
no = 78

78 % 2 = 0

78 / 2 → $\boxed{39}$

```
while(no > 0){
    rem = no%2;
    print(rem);

    no = no/2;
}
```



Powers of 10 → generate at every itr

$1 \times 10^0 = 1$

$0 \times 10^1 = 0$

$1 \times 10^2 = 100$

$1 \times 10^3 = 1000$

1101

8	4	2	1
1	1	0	1

$8 + 4 + 1 = 13$

power = 10^0 = 1 , ans = 0

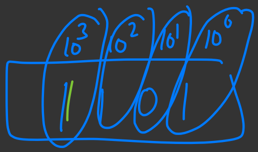
```
while (no > 0){
    last_digit = no%2;
    {
        ans = ans + last_digit × power;
        power = power × 10
    }
    no = no/2;   // update
}
```

→ no&1 ;

correct

work

& Decimal → Binary



ans = 0

$= 10^3 \cdot 1$ = 1000

$+ 10^2 \cdot 1$ + 100

$+ 10^1 \cdot 0$ + 00

$+ 10^0 \cdot 1$ + 1

1101

1 → 1

0 → 00

1 → 100

1 → 1000

1101

power = $10^0 = 1$, ans = 0

```
1101 while (no > 0){
        last_digit = no%2;
          ans = ans + last_digit × power;
          power = power × 10
          no = no/2;      //update
     }
```

w

B

$$p = 1 \rightarrow 10 \rightarrow 100 \rightarrow \underline{1000} \rightarrow 10{,}000$$

$$\boxed{13} \rightarrow 6 \rightarrow 3 \rightarrow 1 \rightarrow 0$$

rem = 1

$$\text{ans} = 0 + 1 \times 1 + 0 + 1 \times 100 + 1 \times 1000$$

$$= 0 + 1 + 100 + 1000$$

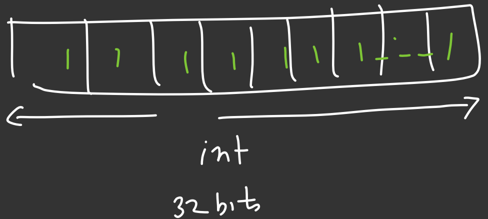
$$\boxed{1101}$$

8 4 2 1

$= 8 + 4 + 1$

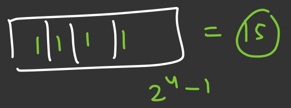
$= 13$

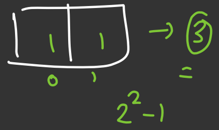
Storage of +ve int



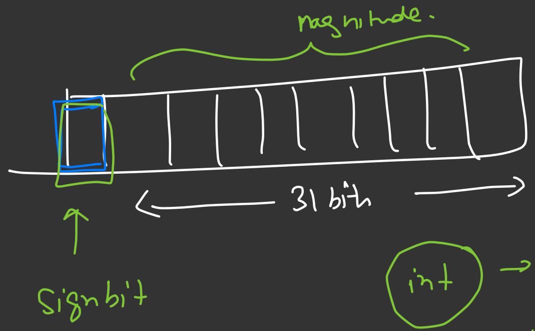
largest → $\cancel{2^{32}-1} \approx 4\times 10^{9}$

$2^{31}-1 \approx 2\times 10^{9}$





Streoge of -ve NO ?



0 ⇒ +ve

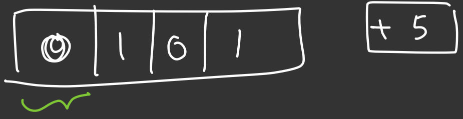
1 ⇒ -ve

NO

int → signed int → front MSB is a sign bit

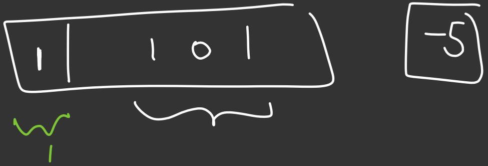
+ve -ve

Ophonal → how negative no's are stored.



4 bit

Signed int



→ But the storage negative is handled different.

↓

2s compliment form

↓ to make subtraction easy & reduce hardware

$\underbrace{6 + 8} \Rightarrow \boxed{14}$

↳ Circuit → $\boxed{\text{Adder}}$ → Addition

→ Subtract

↓

6 + (−8)

5 + (−5)

→ 2S Compliment form

6 − 8 ⇒ −2

↳ Circuit → ~~Subtracter~~

2S Compliment form

↳ Flip all bits of No (positive)

↳ Add 1 to it

← 32 bit →

$\boxed{5}$ ⇒ . . . . $\boxed{0|0|0|0|1|0|1}$ ⇒ $\boxed{5}$

7bit

$\boxed{-5}$ ⇒

1 1 1 1 0 1 0

+ 1

→ $\boxed{1}$ 1 1 1 0 1 1 → $\boxed{-5}$

$\downarrow$

$\boxed{-ve \quad NO}$

$5 - 5 = \textcircled{0}$

$1 + 1 = \boxed{10}$

Subtraction

$5 + (-5)$

$$\begin{array}{cccccccc} & 1 & 1 & 1 & 1 & 1 & & 1 \\ \textcircled{1} & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ + & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} \end{array} \Rightarrow \textcircled{0}$$

Carry discarded

$\leftarrow \quad \rightarrow$

× Powers of 2

$1 \; \boxed{1 \; 1 \; 1 \; 0 \; 1 \; 1} \rightarrow -59$

$\uparrow$

Sign bit == 1

32 bit

$\boxed{\boxed{1} \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1}$

$\uparrow$

Sign bit

-ve

actually how -1 is stored in system.

$-(-x)$

$\downarrow$

$\boxed{+x}$

Negation → 2s compliment

0 0 0 0 0 0 0 0 0 0 0

+ 1

0 0 0 0 0 0 0 0 0 1 => $\boxed{x}$

wrt → problem solving

→ hardware based project

→ SanDisk → firmware engineering

Storage ← Bitwise Operators (fast)

10.30

Logical op   &&, ||, !

Bitwise   &, |, ^, ~, <<, >>

- & → Bitwise And
- | → Bitwise OR
- ^ → Exclusive OR (XOR)
- ~ → Bitwise not
- << → Left Shift
- >> → Right Shift

Bitwise → opeerate directly on bit

AND → integers, char

b1	b2	b1&b2
1	1	1
1	0	0
0	1	0
0	0	0

5 & 7

= 000101
  000111

  000101

= 5

9 & 4

= 0

9 = 1001
4 = 0100

0000 => 0

Use case

Suppose if no is even / odd

→ no & 1 → faster than modulo → no%2 == 0

odd

```
=   1 0 1
  & 0 0 1
  -------
    0 0 1   → 0
```

if last bit is 1

ev

```
6  =   1 1 0
&1 =  &0 0 1
      -------
       0 0 0
```

if (no&1 == 1)
↳ odd

↳ even otherwise

Bitwise OR

b1	b2	b1 | b2
0	0	0
0	1	1
1	0	1
1	1	1

$5|7 = \boxed{7}$

$9|4 = \boxed{13}$

$6|2 = \boxed{6}$

9 => $\boxed{1001}$

OR 100

1101

Algoritms →

⟹ Travelling Salesman

```
 101
 111
 ---
 111
```

```
 1001
 0100
 ----
 1101
```

```
 110
 010
 ---
 110
```

XOR → Exclusive OR ⇒ Exactly one of the bit should be 1.

(Both bits should be different)

^

(directly use it)

b1	b2	b1 ^ b2
0	0	0
0	1	1
1	0	1
1	1	0

→ Exactly 1 bit should be one

$$a \,\&\, b \,\&\, c = a \,\&\, c \,\&\, b = b \,\&\, c \,\&\, a$$

$$\xrightarrow[\text{any order}]{}$$

$$a \mid b \mid c = a \mid c \mid b = c \mid a \mid b \text{ ----}$$

$$a \wedge b \wedge c = a \wedge c \wedge b = c \wedge a \wedge b = c \wedge b \wedge a$$

$5 \wedge 7 = 2$ ✓

$6 \wedge 4 = \boxed{2}$ ✓

$6 \wedge 6 = 0$

$6 \wedge 2 \wedge 6 \wedge 2 = 0$

$$= \underbrace{6 \wedge 6}_{0} \wedge \underbrace{2 \wedge 2}_{0} = \textcircled{0}$$

$$\begin{array}{rl} 5 = & 1\,0\,1 \\ 7 = \textcircled{$\wedge$} & 1\,1\,1 \\ \hline & 0\,1\,0 \Rightarrow \textcircled{2} \\ \hline \end{array}$$

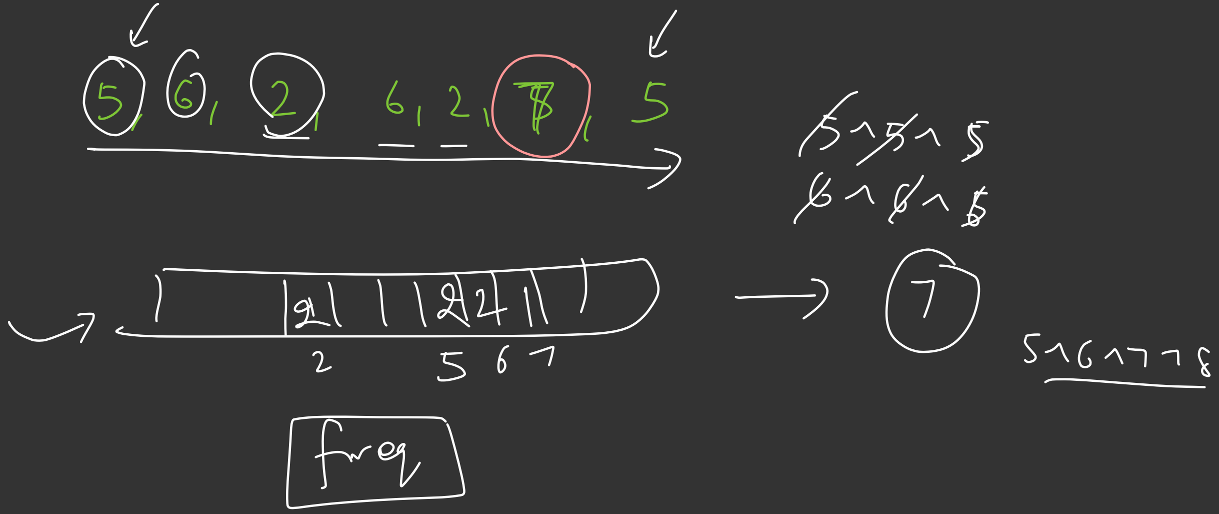
$$\begin{array}{c} 1\,1\,0 \\ 1\,0\,0 \\ \hline 0\,1\,0 \\ \hline \end{array}$$

Q Unique NO

2N + 1 numbers where every no is occ twice except one unique no

find the unique No.

way-1



way-2

$\cancel{5} \wedge \cancel{6} \wedge \cancel{2} \wedge \cancel{6} \wedge \cancel{2} \wedge \boxed{7} \wedge \cancel{5} \wedge 5$

```
res = 0
for(i=0; i<n, i++)(
        res = res^a[i];
    3
```

$5 \cancel{\wedge} \cancel{5} \wedge 5 \wedge \cancel{6} \cancel{\wedge} 6 \wedge 6 \wedge \cancel{7} \wedge \cancel{7} \wedge 7 \wedge \cancel{8} \wedge \cancel{8} \wedge 8$

$= 5 \wedge 6 \wedge 7 \wedge 8$

$=$

```
  0101
   110
 ------
  ----
^  111
 ------
     1
```

Next class + Problems

Bitwise

~

⇒ ~ ↓ Not

>> Right Shift

<< Left Shift

! true

→ Next session

5 → 101

010

6 → 110

$\boxed{\sim 5}$

~1 ⇒ 0

~0 ⇒ 1

$\boxed{\sim 6}$

$\sim$ NO $\Rightarrow$ <u>Flip all bit</u> (inverts)

$\boxed{\sim}\ \boxed{1\ 0\ 1\ 1\ 0\ 0\ 1}$

$\downarrow\ \downarrow\ \downarrow\ \downarrow\ \downarrow\ \downarrow\ \downarrow$

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

$\textcircled{1} \wedge 0 = \textcircled{1}$

$\textcircled{0} \wedge 0 = 0$

$\sim 1 \Rightarrow 0$

$\sim 0 \Rightarrow 1$