Single Number → ans = 0

↓ ans ^= a[i] ✓

Dictionary

XOR

$a \wedge 1 = \begin{cases} a-1 \\ a+1 \end{cases}$

$a \mid 1 = \begin{cases} a \\ a+1 \end{cases}$

$a \& 1 \begin{cases} 0 \\ 1 \end{cases}$

$\boxed{\begin{matrix} 1 \wedge 1 = 0 \\ 0 \wedge 0 = 0 \end{matrix}}$

$\boxed{a \wedge a = 0}$

$a \wedge 0 = a$

$\underbrace{a \wedge a}_{0} \wedge \underbrace{b \wedge b}_{0} \wedge \underbrace{c \wedge c}_{0} \wedge d$

$\boxed{\& \quad \mid \quad \wedge}$ ⇒ $\boxed{O(1)}$ ←

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

13 & 17

a + b

+
*
/
} slower min 5 clock cycles

⇒ bitwise operation → 1 clock cycle

$(11)_{10} \Rightarrow (1011)_2$

(bit positions 4 3 2 1 over 1 0 1 1)

check if 2nd bit is ON/OFF

$\boxed{0010}$

11 & (2) = 2 ← non zero value
bit is ON

9 & 2 = 0 ← else bit is OFF

100 if 4th bit is on OR off

& 001000

AND with 8

100 & 2 (2×2) ⇒ 2nd bit 000010

& 8 ⇒ 4th bit 001000

& 16 ⇒ 5th bit 010000

(10000) (positions 5 4 3 2 1)

& (2^{K-1}) = Kth bit is ON or OFF

$(11)_{10} = (1011)_2$ ←

$*2$

$(22)_{10} = (10110)_2$

$*2$

$(44)_{10} = (101100)_2$

$11 = 8 + 2 + 1$

$= \boxed{2^3 + 2^1 + 2^0}$

$(22) = 11 * 2$

$22 = 11 * 2 = 2(2^3 + 2^1 + 2^0) =$ 1 0 1 1

$= \boxed{2^4 + 2^2 + 2^1}$

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

We observe that when we multiply a number by 2, the bits in binary representation shift towards left.

(Last added bit is zero)

$\Rightarrow$ << Left shift operator

$\boxed{a << 1}$

print(10 << 1)

$(1010)_2$

$(10100)_2$

20

$\underline{a << n}$

$a << 1 \; = \; a*2 \; = a * pow(2,1)$

$a << 2 \; == \; a*4 \; = a * pow(2,2)$

$a << n \; = \; \ldots \; \boxed{a * power(2,n)}$

Right shift operator

$\underline{a >> 1}$

$11 >> 1 \Rightarrow \boxed{1011} \; 11$

$\Downarrow$

$\boxed{0101} \; 5$

Integer division by 2

$$\boxed{11/2 == 11 >> 1}$$

$11 = 8 + 2 + 1$

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

$= 2^3/2 + 2^1/2 + 2^0/2$

$\boxed{2^2 + 2^0} + \cancel{2^0/2}$ $\cancel{0.5}$

#5

check if 6th bit is ON/OFF

in numb 997 = x

6 5 4 3 2 1

$2^{(6-1)}$

Right shift operator

k

$x \,\&\, (1 << (k-1)) \;!= 0$

22

10110 &1

$(x >> (k-1)) \,\&\, 1$

2^6

10110

& 01000 == 1<<3

10110

1011

101

10

3 time right shifting

$(x >> (k-1) \,\&\, 1)$

10 & 1

1010

$\downarrow$ >> 1

0101

$\downarrow$ >> 1

4

0010

>> 1

$\downarrow$

10 >> 4 == 0

0001

$\downarrow$

$\boxed{50 >> 2}$

0000 ✓

11001~~0~~ == 50

$\boxed{11001}$ ⇒ 25 ✓

Integer division by 2 in right shift operator

$\boxed{1100}$ ⇒ 12

Q find no. of bits switched ON in number x

Binary | left shift | Right shift

$\_ \_ \_ \_$ ⇒ 4 bits ⇒ different no.s can we store

$= 2^4$

$\_ \_ \_ \_ \_ \_ \_ \_ = 2^8 = \boxed{0 \text{ to } 255}$

32 bits $= 2^{32}$ (0 to $(2^{32}-1)$)

$=$ 0 to 10^9

$\approx 4 \times 10^9$

18 digits 64 bits $\boxed{2^{64}}$

10^{18}

64 bits to cover

Q Unset the last set bit :

50 = $\boxed{110010}$ 110010

$\Downarrow$

48   $\boxed{110000}$

① find i where i^{th} bit ON

② - $\boxed{x = x \wedge (1 \ll i)}$

```
i = 0
while True:
    if x & (1<<i) != 0
        break.
    i += 1
```

$$\boxed{x = x \wedge (1 \ll i)}$$

return x

$$
\begin{array}{r}
110010 \\
-\quad 1 \\
\hline
110001
\end{array}
$$

$$
\begin{array}{r}
1100011000 \\
-\quad 1 \\
\hline
1100010111
\end{array}
$$

$\Downarrow$

1100010000

$$
\begin{array}{r}
10000 \\
-\quad 1 \\
\hline
9999
\end{array}
$$

$N = \boxed{N \,\&\, (N-1)}$

unsets the last set bit

110010

$\Downarrow$

110000

$\Downarrow$

100000

$\Downarrow$

0

3 bits

$\boxed{N\&(N-1)}$

N & N-1

signed bits

⇓

How do we represent negative no.s

$2^3 = -2^3 \Rightarrow 8$

-ve

_ _ _ _

↓

ON

1

Then number is negative

This bit represents -ve magnitude

2^4 numbers

$= \boxed{0 - 15} \Rightarrow$ +ve no.s are represented

ONLY 16 numbers can be represented in 4 bits.

$\boxed{-8 \text{ to } 7} \Rightarrow$ 16 numbers

2's complements

0	→	0000
1	→	0001
:		
:		
7		0111

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

-8 ⇒	$\boxed{1000}$
-7 =	$\boxed{1001}$
-6 =	1010
-5	1011
-4	$\boxed{1100}$
-3	1101
-2	1110
-1	$\boxed{1111}$

signed Data types

$-2^3 + 0 + 0 + 0 = -8$

1 0 0 1

$-2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$-2^3 + 2^0$

$-8 + 1 = -7$

$\boxed{-2^7}$ $2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$2^7 - 1$

8   8 bits $\boxed{0 - 255}$

$\boxed{-128 \text{ to } 127}$

-1 => 8 bits ?

-2^7

GP

$$-2^7 + \boxed{2^6 + 2^5 \cdots 2^0}$$

$$\cancel{-2^7} + \cancel{2^7} - 1$$

-3 binary representation

2's complements

$-3 = 00000011$

Complement

11111100

$+ \quad 1$

11111101 (-3)

chr( )

char( )

int('01', 2)

$(\ )_{10}$

int( )

ord( )

Algo 2's complement.

1's complement

1. Toggle all bits of its positive magnitude representation
2. Add

$-8 = 1000$

0111

1000

$-4 = 0100$

0011

$+1$ $\boxed{1100}$

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

$i + j \Rightarrow$ constant

[[ ]
[ ]
[ ]
]

1 TC → code you have written

N Look at the no. of data values

10^8 → iterations per sec = 2 secs

2×10^8

N, N^2 <

$N \log N$

$N \sim 10^3$

N^2

$10^6 < 10^8$