Bit Manipulation - I

Agenda

1 - Number System

- Binary Number System
- Conversion

2 - Add 2 Binary Numbers.

3 - Bitwise Operators : ^, &, |, ~, <<, >>

^ : XOR

<<, >> : Next session

4 - Properties

5 - Problems 2-3

⇒ Representation

Expand $(197) = 1 * 100 + 9 * 10 + 7 * 1 = 1 \cdot 10^2 + 9 \cdot 10^1 + 7 \cdot 10^0$

1, 0

System	Digits	Count # digits.
✓ Decimal	0 – 9	10
Hexadecimal	0 – 9, A, B, C, D, E, F	16
Octal	0 – 7	8
✓ Binary	0 – 1	2

$(197)_{10}$

$(N)_B$

base

Value associated

Binary ⟶ Decimal

Binary number

$(10101)_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4$

$= 1 + 0 + 4 + 0 + 16$

$= \boxed{21}$

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

binary representation ⟵ given

Decimal → Binary

res = [1, 9, 7]

$\overset{2\ 1\ 0}{197} = 1 \cdot 10^2 + 9 \cdot 10^1 + 7 \cdot 10^0$

$\boxed{N}$

1) $\frac{197}{100} = 1$ $\quad N/10^2$

$\underbrace{197 \% 100}{} = 97$ $\quad N = N \% 10^2$

2) $97/10 = 9$ $\quad N/10^1$

$97 \% 10 = 7$ $\quad N = N \% 10$

3) $7/1 = 7$ $\quad N/1$

$7 \% 1 = \boxed{0}$ $\quad N = N \% 1$

[1, 9, 7]

Pseudo-code

```
res = []
K = len of the number - 1
while N > 0 :
    res.append(N // (10**K))
    N = N % (10**K)
    K -= 1
```

process

8 → binary

Change 10 by 2.

		rem
2	8	0
2	4	0
2	2	0
2	1	1
	0	

$(1000)_2$

len = 4

		rem
10	197	7
10	19	9
10	1	1
	0	

$(197)_{10}$

len = 3

$\frac{N}{2^K} = 1$

$\Rightarrow 2^K = N$

$\Rightarrow K = \log_2 N$

$2 \overline{)\,1\,(}\ 0$

1

quotient

$\boxed{1} = 2 * 0 + 1$

divison dividend remainder

		rem
2	10	0
2	5	1
2	2	0
2	1	1
	0	

bottom-up

$$\boxed{(1010)_2}$$

$$\begin{array}{r} 2 \\ 2\,\overline{)\,5} \\ \underline{4} \\ \underline{1} \end{array}$$

q

a

b

r

$$5 = 2 * 2 + 1$$

$$\underline{a} = \underline{b} \cdot q + r$$

Q 1 Given 2 binary strings, add them up.

$$\begin{array}{r} 00101 \Rightarrow 5 \\ 10010 \Rightarrow 18 \\ \hline 10111 \Rightarrow 23 \end{array}$$

Decimal

$$\begin{array}{r} \text{sum}/10 \\ 1\,1\,1 \\ 1234 \\ 1789 \\ \hline 3023 \\ \text{sum}\%10 \end{array}$$

$\boxed{1}$ $(8+5)/10$

$$\begin{array}{r} 5 \\ 18 \\ \hline 23 \end{array}$$

$(8+5)\%10$

$1\%2 = 1$

$1/2 = 0$

$\boxed{2/2 = 1}$

$\boxed{3/2 = 1}$

$\boxed{4/2 = 2}$

$c = (a+b)/2$

$$\begin{array}{rr} & 1\,1\,1\,1\,1\,1\,1\,1 \leftarrow \text{carry} \\ & 0\,0\,1\,1\,1\,1\,1\,1 \rightarrow a \\ + & 0\,1\,0\,1\,0\,1\,1\,1 \rightarrow b \\ \hline & 1\,0\,0\,1\,0\,1\,1\,0 \leftarrow \text{digit} \end{array}$$

$(1+1)\%2 = 0$

$(1+1)/2 = 1$

$d = (a+b)\%2$

TC: $O(\max(m,n))$ m = len of 1st num

SC: n = len of 2nd num.

Quiz

$$\begin{array}{r} {\scriptstyle 1\ 1} \\ (0110)_2 \\ (0110)_2 \\ \hline (1100)_2 \end{array} \quad \begin{array}{l} \\ = 6 \\ = 6 \\ = 12 \end{array}$$

⇒ Bitwise Operators

0 → unset bit
1 → set bit.

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ \boxed{2^3} & 2^2 & 2^1 & 2^0 \end{array}$$

MSB, LSB

n bits $= 4$

$8 + 4 + 2 + 1 = 15 = 2^4 - 1.$

$2^n - 1$

$$= 2^{n-1} + 2^{n-2} + 2^{n-3} + \dots + 2^1 + 2^0 \quad (GP)$$

$$= 1 + 2 + 2^2 + \dots + 2^{n-1}$$

$$a + ar + ar^2 + \dots + ar^{n-1}$$

$$= \frac{a(r^n - 1)}{r - 1}$$

$$= \frac{1(2^n - 1)}{2 - 1} = \boxed{2^n - 1}$$

Q) For a number n how many max bits are needed to store it?

$$7 \rightarrow \overset{2\,1\,0}{111} \qquad 2^2+2^1+2^0 = 4+2+1 = 7$$

$$15 \rightarrow \overset{3\,2\,1\,0}{1111} \qquad 2^3+2^2+2^1+2^0 = 8+4+2+1 = 15$$

15

↓

7

↓

3

↓

1

↓

0

4 steps needed to reduce to zero

$$N \rightarrow N/2 \rightarrow N/2^2 \rightarrow \text{---} \rightarrow \underbrace{\boxed{1}}_{N/2^K} \rightarrow 0$$

$$\frac{N}{2^K} = 1$$

$$\Rightarrow K = \log N$$

$O(\log_2 N)$ bits needed.

Applications of Bitwise Operators

1 — checking if a num is odd/even.   more efficient

2 - encryption algorithms

3 - compression algorithms

4 — Parity of number → count of set bits even
↳ count of set bits odd

⇓

Error Correction Codes.

Hamming distance

Networking Algorithm

Bitwise Operators : how will they work

bit by bit on numbers.

1. XOR operator (^)

$a = 0\,1\,0\,1\,0\,1\,0\,\boxed{0}$

$b = 1\,1\,1\,1\,1\,0\,1\,\boxed{1}$ ^

$a \wedge b = 1\,0\,1\,0\,1\,1\,1\,1$

x_1	x_2	$x_1 \wedge x_2$
0	0	0
0	1	1
1	1	0
1	0	1

same bits $0 \wedge 0 = 0$

$1 \wedge 1 = 0$

diff. bits $0 \wedge 1 = 1$

$1 \wedge 0 = 1$

Property

$a \wedge 0$

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

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

$1 \wedge 0 = 1$

$0 \wedge 0 = 0$

and (&)	or (|)
ampersand	pipe

a 101011

b 000011

&

a&b = 000011

a 101011

b 000011

|

a|b = 101011

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

both have to be set

either one of a & b can be set.

	3	2	1	0	
10:	1	0	1	0	← 10
14:	1	1	1	0	← 14

10 → 8 + 2

14 → 8 + 4 + 2

10 ^ 14   0 1 0 0 → 4

10 & 14   1 0 1 0 → 10

10 | 14   1 1 1 0 → 14

28

NOT operator

~

tilde

a: 1 0 1 0 1 1 1 0

~a: 0 1 0 1 0 0 0 1

a	~a
0	1
1	0

set → unset

unset → bit

Toggle every bit

Properties

$1 \wedge 0 = 1$

$0 \wedge 0 = 0$

XOR

1. $a \wedge 0 = a$ retaining.

2. $\boxed{a \wedge a} = 0$ cancelling

$1 \wedge 1 = 0$

$0 \wedge 0 = 0$

$$\boxed{\begin{array}{l} a \,\&\, a = a \\ a \,|\, a = a \end{array}}$$

$$\begin{array}{r} 101101 \\ \wedge\ 101101 \\ \hline 000000 \end{array}$$

Other

3. $a \,\&\, 0 = 0$

$$\begin{array}{r} 10110 \\ 00000 \\ \hline 00000 \end{array}$$

4. $a \,|\, 0 = a$

$$\begin{array}{r} 10110 \\ |\ \ 00000 \\ \hline 10110 \end{array}$$

---

5. $a \,\&\, 1 = \begin{cases} 0 \\ 1 \end{cases}$

$\Downarrow$

00001

$\Downarrow$

gives us the last bit of a (LSB)

0 ← LSB

$$\begin{array}{rl} a = & 101101 \\ 1 = & 000001 \\ \hline a \,\&\, 1 = & 000001 \end{array} \Rightarrow \text{odd}$$

$$\begin{array}{rl} a = & 101100 \\ 1 = & 000001 \\ \hline a \,\&\, 1 = & 000000 \end{array} \Rightarrow \text{even}$$

$a_3 \quad a_2 \quad a_1 \quad a_0$

$= [a_0 \cdot 2^0] + a_1 \cdot 2^1 + a_2 \cdot 2^2 + a_3 \cdot 2^3$

$= [a_0] + 2[a_1 + a_2 \cdot 2 + a_3 \cdot 2^2]$

$0 \quad 1$

odd
$2x+1$
even
$2y+0$

---

6. $a \mid 1 = \begin{cases} a+1, & \text{even} \\ a, & \text{odd} \end{cases}$

$$\begin{array}{r} 1010110\,1 \\ 0000000\,1 \\ \hline 1010110\,\boxed{1} \end{array} \; \Big\} \text{ odd} \Rightarrow a$$

$a \mid 0 = a$

$$\begin{array}{rr} a & 1010110\,0 \\ 1 & 0000000\,1 \\ \hline \end{array} \; \Big\} \text{ even}$$

$a \mid 1 = \underbrace{1010110}\,\boxed{1} \Rightarrow \boxed{a+1}$

$= a+1.$

7. $a \wedge 1$: Try as HW.

(>2 numbers)

Commutative and Associative

$a \wedge b = b \wedge a$	$(a \wedge b) \wedge c = a \wedge (b \wedge c)$
$a \& b = b \& a$	$(a \& b) \& c = a \& (b \& c)$
$a \mid b = b \mid a$	$(a \mid b) \mid c = a \mid (b \mid c)$

Q2 You are given N numbers where every number appears twice except one number. (appear just once)

Find that one number.

arr = [1, 2, 2, 7, 6, 5, 5, 6, 1]

Brute Force : $O(N^2)$ For every number

loop over all other numbers

check if cnt == 2.

Hashing ⇐ Dictionary $O(N)$ time

SC : $O(N)$

Sort TC: $O(N \log N)$

Achieve : $O(N)$ time & $O(1)$ space.

Idea

$$a \wedge b = K$$

XOR a both sides

$$(a \wedge b) \wedge a = (\underline{a \wedge a}) \wedge b = b = K \wedge a$$

$$a \wedge a \Rightarrow 0$$

$$0 \wedge b = b$$

$$\Rightarrow \boxed{b = K \wedge a}$$

XOR b on both sides.

$$a \wedge (b \wedge b) = K \wedge b$$

$$a \wedge 0 = K \wedge b$$

$$\Rightarrow \boxed{a = K \wedge b}$$

XOR all the nums

$$\Rightarrow 1 \wedge 2 \wedge 2 \wedge 7 \wedge 6 \wedge 5 \wedge 5 \wedge 6 \wedge 1$$

$$(1 \wedge 1) \wedge (2 \wedge 2) \wedge (6 \wedge 6) \wedge (5 \wedge 5) \wedge 7$$

$$= 0 \wedge 0 \wedge 0 \wedge 0 \wedge 7$$

$$= \boxed{7}$$

$a \wedge b \Rightarrow O(1)$

$a \,\&\, b \Rightarrow O(1)$

$a \,|\, b \Rightarrow O(1)$

Fixed-length numbers

bits are constant

How to find XOR?

$a + 0 = \boxed{a}$

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

sum

```
sum = 0
for i in range(0, n):
    sum += arr[i]
```

xor

n iterations

```
xor = 0
for i in range(0, n):
    xor = xor ^ arr(i)    # xor ^= arr[i]
```

TC: $O(N)$

SC: $O(1)$

×