Bit Manipulation - II

00100 → 4 (1<<2) [2^2 = 1<<2]

Power of left shift

OR operator

$$N = 45 \rightarrow \begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{r|cccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ \text{OR } (1<<2) & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline & 1 & 0 & 1 & \textcircled{1} & 0 & 1 \end{array} \rightarrow 45$$

$$\begin{array}{r|cccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ \text{OR } (1<<4) & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline & 1 & \textcircled{1} & 1 & 1 & 0 & 1 \end{array} \rightarrow 61$$

[61 - 45 = ⑯]

$$\boxed{N \mid (1<<i) = \text{set } i^{th} \text{ bit in N if it is unset (means 0) else NO CHANGE}}$$

$= N + (1<<i)$ if i^{th} bit is unset

$= N$ else

$1<<i \Rightarrow 2^i \Rightarrow \underset{}{0}\ 0\ 0\ \underset{i^{th}}{1}\ 0\ 0\ 0\ 0$

XOR operator

$$\begin{array}{r|cccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ \text{XOR } (1<<2) & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \begin{array}{l} = 45 \\ \\ = 41 \end{array}$$

$$\begin{array}{r|cccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ (1<<4) & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline & 1 & 1 & 1 & 1 & 0 & 1 \end{array} = 61$$

$$\boxed{N \wedge (1<<i) \rightarrow \text{flips/ toggles } i^{th} \text{ bit}}$$

$$N \wedge (1<<i) = \begin{cases} N + (1<<i) & \text{if } i^{th} \text{ bit is unset} \\ N - (1<<i) & \text{if } i^{th} \text{ bit is set} \end{cases}$$

AND operator

$$\begin{array}{r|cccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ \text{AND } (1<<2) & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \rightarrow 2^2 = 4$$

$$\begin{array}{r|cccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ \text{AND } (1<<4) & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$N \& (1<<i) \begin{cases} \rightarrow (1<<i) & \text{if } i^{th} \text{ bit is set} \\ \rightarrow 0 & \text{ELSE} \end{cases}$$

Question 1

Unset i^{th} bit of a number if it is set,
else DO NOTHING.

$N = 45 \quad 1\ 0\ 1\ 1\ 0\ 1$

if $i=2 \Rightarrow 1\ 0\ 1\ 0\ 0\ 1$

if $i=4 \Rightarrow 1\ 0\ 1\ 1\ 0\ 1$

Since, we don't know that i^{th} bit is set/ unset we can't toggle. However, we can set and then toggle.

Code

`x = N|(1<<i)` → this will set i^{th} bit

`ans = x^(1<<i)` → this will toggle i^{th} bit which means unset i^{th} bit

Alternatively

```
if (checkBit(N,i))
    N = N^(1<<i)
```

→ `N|(1<<i) == N`

→ `N&(1<<i) == (1<<i)`

→ `N&(1<<i) != 0`

Question 2

Check if i^{th} bit is set?

1. `N|(1<<i) == N`
2. `N&(1<<i) == (1<<i)`
3. `N&(1<<i) != 0`
4. `N^(1<<i) < N`

`N^(1<<i)` ↗ `N+(1<<i)` if i^{th} bit is unset
↘ `N-(1<<i)` if set

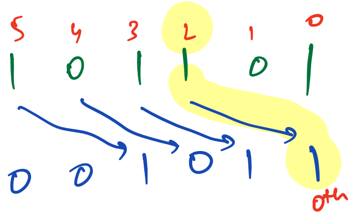
`N&1` → 1 if 0^{th} bit is set (odd no.)
→ 0 else (even no.)

N=45

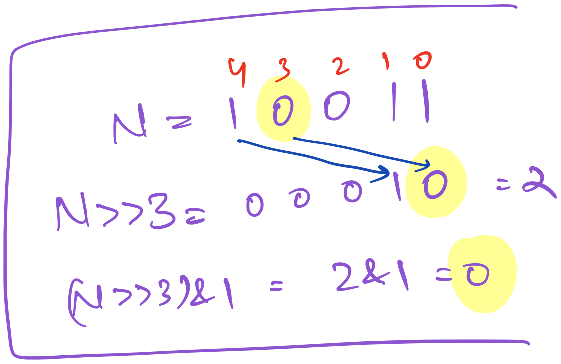
i=2

(N>>i)

(N>>2)



S. (N>>i) & 1 == 1



Question 3

Count the number of set bits in N.

int → 32 bits , long → 64 bits

```
count=0
for (i=0; i<32; ++i) {
    if((N>>i)&1)
        ++count
}
```

TC: O(1)

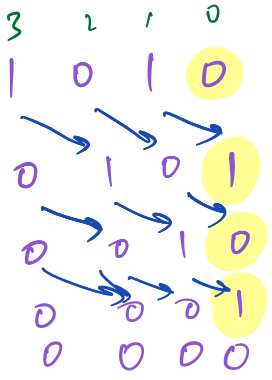
N=10

N>>1

N>>2

N>>3

N>>4



N&1 = 0 ((N>>0)&1)

(N>>1)&1 = 1

(N>>2)&1 = 0

(N>>3)&1 = 1

```
count =0
while (N>0) {
    if (N&1)
        ++count
    N = N>>1
}
```

$10 \& 1 = 0$

$5 \& 1 = 1$

Negative numbers

$(-45)_{10} = (?)_2$

integer → 32 bits

MSB - most significant bit

31	30	29	28	..	..	..	2	1	0

$$2^{31} > \underbrace{2^{30} + 2^{29} + 2^{28} + \dots \dots 2^2 + 2^1 + 2^0}_{31 \text{ terms}}$$

GP $a = 2^0$ $r=2$ $n=31$

$$\text{sum} = a\left(\frac{r^n - 1}{r-1}\right) = 1\left(\frac{2^{31}-1}{2-1}\right)$$

$$= 2^{31} - 1$$

for explaining negative number, we'll use 8 bit numbers.

	7	6	5	4	3	2	1	0
45 →	0	0	1	0	1	1	0	1

flip all bits

1's complement of 45 →	1	1	0	1	0	0	1	0
+1 →	0	0	0	0	0	0	1	

2's complement of 45 → 1 1 0 1 0 0 1 1

$2^7 + 2^6 + 2^4 + 2^1 + 2^0 = 128+64+16+2+1$

$= 211 = -45$

$-2^7 + 2^6 + 2^4 + 2^1 + 2^0 = -128 + 64 + 16 + 2 + 1$

$= -45$

1 1 1 0 0 1 1 1

$-2^7 \quad 2^6 \quad 2^5 \quad 2^2 \quad 2^1 \quad 2^0$

$= -128+64+32+4+2+1$

$= -25$

Before negative no.

Min Val : 00 00 00 00 = 0

Max Val : 1 1 1 1 1 1 1 1 = 255

8 bit unsigned int

After negative no.

Min val : 1 0000000 $= -2^7 = -128$

Max val : 0 1111111 $= 127$

$2^7 - 1$

normal 8 bit int (signed)

for 32 bit integers

Min val : $-2^{31} = -2147483648 \approx -2\times10^{9}$

Max val : $2^{31}-1 = 2147483647 \approx 2\times10^{9}$

for 64 bit integer (long)

Min val : $-2^{63} \approx -9\times10^{18}$

Max val : $2^{63}-1 \approx 9\times10^{18}$

Question

Calculate sum of all elements in an array.

constraints : $1 <= N <= 10^5$

$1 <= A[i] <= 10^6$

```
~~int~~ long sum=0
for i=0 to N-1
    sum += A[i]
return sum
```

$A = [10^6 \; 10^6 \; .... \; 10^6]$ $N = 10^5$

$sum = 10^6 \times 10^5 = 10^{11} > 2\times10^9$

constraints → TLE

↳ overflow

Multiple 2 numbers

int a,b $\quad$ $a, b <= 2 \times 10^9$

find a×b ?

$a \times b <= 4 \times 10^8$

we need long

int ans = a×b; ✗

long ans = $\boxed{a \times b}$; ✗

↳ overflow will happen at multiply step

long ans = (long)(a×b); ✗

long ans = $\underbrace{\text{long}(a)}_{\text{long}} \times \underbrace{b}_{\text{int}}$; ✓

long × int = long

CPU

store a

store b

overflow step → $\boxed{\text{MUL } a, b \text{ to temp}}$

COPY temp to ans

long ans = a;

ans *= b; ✓

long ans = long(a) × long(b); ✓

Subtract 2 binary numbers [8 bit]

45 - 12

45 + (-12)

↳ 2's complement of 12

12= 00001100

1's comple = 11110011

2's = 11110100

$$\begin{array}{rcccccccc}
 & 1 & 1 & 1 & 1 & 1 & & & \\
45 = & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
-12 = & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
\hline
① & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}$$

discard

↳ $2^5+1 = 33$

(45-12)