(Saturday)
↳
Optional

BIT Manip - 3

↳ Prob Solving Session
↳ Doubt session

HW probs: Loco PSPs

* Given a binary no. with only 1 bit set

$$\begin{array}{ccccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

$(0000100)_2 \rightarrow 2^2 = 4$ (1<<2)

$$\begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

$(010000)_2 \rightarrow 2^4 = 16$ (1<<4)

Set = 1
Unset = 0

⊛ Power of left shift:

$N = 45 \rightarrow$

$$\begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

OR

$$\begin{array}{l|cccccc} & 5 & 4 & 3 & 2 & 1 & 0 \\ (45) & 1 & 0 & 1 & 1 & 0 & 1 \\ (1<<2) & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 45 \leftarrow & 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{l|cccccc|l} & 5 & 4 & 3 & 2 & 1 & 0 & \\ (45) & 1 & 0 & 1 & 1 & 0 & 1 & \rightarrow 45 \\ (1<<4) & 0 & 1 & 0 & 0 & 0 & 0 & \rightarrow 16 \\ \hline & 1 & 1 & 1 & 1 & 0 & 1 & \\ \end{array}$$

↳ 61

Obs: N | (1<<i)
↳ Setting the i^{th} bit, if it is unset
→ N, if i^{th} bit is set
→ N + (1<<i), if i^{th} bit is unset

XOR

	5	4	3	2	1	0	
(45)	1	0	1	1	0	1	(45)
(1<<2)	0	0	0	1	0	0	(4)
41 ←	1	0	1	0	0	1	41

	5	4	3	2	1	0	
(45)	1	0	1	1	0	1	(45)
(1<<4)	0	1	0	0	0	0	(16)
	1	1	1	1	0	1	61

↳ 61

Obs: N^(1<<i) : flips the i^{th} bit

↳ toggles

N^(1<<i) → N+(1<<i), i^{th} bit is unset

→ N-(1<<i), i^{th} bit is set

AND

N=45

	5	4	3	2	1	0	
	1	0	1	1	0	1	(45)
(1<<2)	0	0	0	1	0	0	(4)
	0	0	0	1	0	0	

↳ 4

5	4	3	2	1	0	
1	0	1	1	0	1	(45)
0	1	0	0	0	0	(16)
0	0	0	0	0	0	→ 0

Obs: N&(1<<i) : checks if i^{th} bit is set

→ 1<<i, if i^{th} bit is set.

→ 0, if i^{th} bit is unset.

Qn: Given a no. N. Unset ith bit, if it is set else, no change.

N=45

$$\begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}, \; i = 2 \Rightarrow \begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{array} = 41$$

$$\begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}, \; i = 4 \Rightarrow \begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

```
if(CheckBit(N,i)) {
        N = N ^ (1<<i)
}
else → No change
```

>0

① $N \,\&\, (1<<i) == (1<<i)$

② $N \,|\, (1<<i) == N$

③ $N \wedge (1<<i) < N$

Qn: Check if ith bit is set.

$N = 45$ $\quad$ $\begin{matrix} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{matrix}$ $\quad$ $i = 2$ → true

$N = 45$ $\quad$ $\begin{matrix} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{matrix}$ $\quad$ $i = 4$ → false

>0

① $N \,\&\, (1<<i) == \underline{(1<<i)}$

② $N \mid (1<<i) == N$

③ $N \wedge (1<<i) < N$

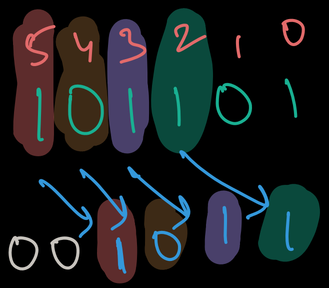
Left shift operator is banned.
↳ not allowed.

pow(2,i) X

$\underline{N \,\&\, 1}$ → 1, 0th bit is set (odd no)
↳ 0, 0th bit is unset (even no)

$\underline{>>}$

$N = 45$
$i = 2$
$\underline{N >> i}$



④ $(N >> i) \,\&\, 1 == 1$

Qn: Count the no. of set bits in N

N=45 101101 ans = 4

int → 32 bits , long → 64 bits

```
ans = 0
for(i=0; i < 32 or 64; i++) {
        if(checkBit(N,i)) {
                ans++
        }
}
return ans
```

✓ works → if datatype of N = fixed

O(1)

O(N)

O(log N)

↓ if its arbitary

10 → long

	3	2	1	0	
N= 10	1	0	1	0	(10)
N>>1	0	1	0	1	(5)
N>>2	0	0	1	0	(2)
N>>3	0	0	0	1	(1)
N >>4	0	0	0	0	→ 0

ans = ans + (N&1)

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

```
ans = 0
while(N>0) {
    ans = ans+(N&1)
    N = N>>1
}
return ans
```

TC: O(log N)

SC: O(1)

N : _ _ _ _ _ _ _ _ - - - - _

int: $2^{32} \rightarrow \log_2(2^{32}) = \boxed{32 \text{ bits}}$

For any no N, there are roughly ≈ $\log_2 N$ bits

Break: till → 8:35am

$$\begin{array}{cccccccc} & & & x & & & y & \\ A & B & A \wedge B & \underline{A|B} & \sim A & \sim B & \underline{\sim A|\sim B} & x \,\&\, y \end{array}$$

Negative Numbers

$(-45)_{10} \rightarrow (??)_2$

int → 32 bits

$\underline{\ \ }$ $\underline{\ \ }$ $\underline{\ \ }$ $\underline{\ \ }$ $\underline{\ \ }$ $\underline{\ \ }$ . . . . $\underline{\ \ }$ $\underline{\ \ }$ $\underline{\ \ }$ $\underline{\ \ }$

31 | 30 29 28 27 26 . . . 3 2 1 0

$$2^{30} + 2^{29} + 2^{28} + \dots + 2^2 + 2^1 + 2^0$$

GP series

$$2^{31} > \frac{2^0(2^{31}-1)}{2-1} = 2^{31}-1$$

MSB: Most Significant Bit

↳ 0 : +ve number

↳ 1 : -ve number

8bit no.

7 6 5 4 3 2 1 0

N = 45 ⇒ 0 0 1 0 1 1 0 1

① flip all bits ⇒ 1 1 0 1 0 0 1 0 [1s complement]

② add 1 (+) 0 0 0 0 0 0 0 1 [2s complement]

1 1 0 1 0 0 1 1 ⇒ −45 in binary

7 6 5 4 3 2 1 0

$-2^7 + 2^6 + 2^4 + 2^1 + 2^0$

$= -128 + 64 + 16 + 2 + 1$

$= \boxed{-45}$

Qn: −12 in binary ?

7 6 5 4 3 2 1 0

12: 0 0 0 0 1 1 0 0

1 1

① flip all bits 1 1 1 1 0 0 1 1

② add 1 (+) 0 0 0 0 0 0 0 1

1 1 1 1 0 1 0 0

$-2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^2$

$= -128 + \underline{64 + 32} + \underline{16 + 4}$

$= -128 + 116$

$= \boxed{-12}$

(*) Min no. in 8 bit system:

$$\begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$= -128$

(*) Max no. in 8 bit:

$$\begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

$= 127$

Range of ints

Min no.:

$$\begin{array}{cccccc} 31 & 30 & 29 & & 1 & 0 \\ 1 & 0 & 0 & \ldots\ldots & 0 & 0 \end{array}$$

-2^{31} : -2147483648

$\approx -2 \times 10^9$

Max no:

$$\begin{array}{cccccc} 31 & 30 & 29 & & 1 & 0 \\ 0 & 1 & 1 & \ldots\ldots & 1 & 1 \end{array}$$

$2^{31} - 1$: 2147483647

$\approx 2 \times 10^9$

Range of long (64 bits)

Min no:

$$\begin{array}{ccccccc} 63 & 62 & 61 & 60 & & 1 & 0 \\ 1 & 0 & 0 & 0 & \ldots\ldots & 0 & 0 \end{array}$$

$= -2^{63}$: -9,223,372,036,854,775808

$\approx -9 \times 10^{18}$

Max no:

$$\begin{array}{ccccccc} 63 & 62 & 61 & 60 & & 1 & 0 \\ 0 & 1 & 1 & 1 & \ldots\ldots & 1 & 1 \end{array}$$

$= 2^{63} - 1$: 9,223,372,036,854,775807

$\approx 9 \times 10^{18}$

Qn: Calc sum of all array elements int a[].

```
~~int~~ → long sum=0
for(i=0; i<N; i++) {
        sum= sum+a[i]
}
return sum
```

Constraints:

$1 <= N <= 10^5$

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

Case: $A = [10^6 \quad 10^6 \quad 10^6 \quad ...]$ $N = 10^5$

$sum = 10^6 \times 10^5$

$= 10^{11}$

↳ overflow.

Constraint → TLE

↳ overflow

Qn: Given 2 ints, a & b. Return their product.

```
int ans = a*b
return ans
```
✗ overflow ←

$a <= 2 \times 10^9$

$b <= 2 \times 10^9$

$a * b <= 4 \times 10^{18}$

⇓

```
long ans = a*b
return ans
```
✗ → Causes overflow ← a*b is done first then it is stored in ans

⇓

```
long ans = long(a*b)
return ans
```
✗ ↳ overflow

⇓

```
long ans = long(a)*b
return ans
```
✓

long * int = long

long ans = ((long) a) * b // in java

(*) Multiply w/o typecasting?

```
long ans = a
ans = ans*b
return ans
```
✓

```
long ans = 1L * a*b
return ans
```
✓

1L: 1 rep as a long

(*) How to subtract binary numbers?

Q 45 - 12

↳ 45 + (-12)

8 bit memory

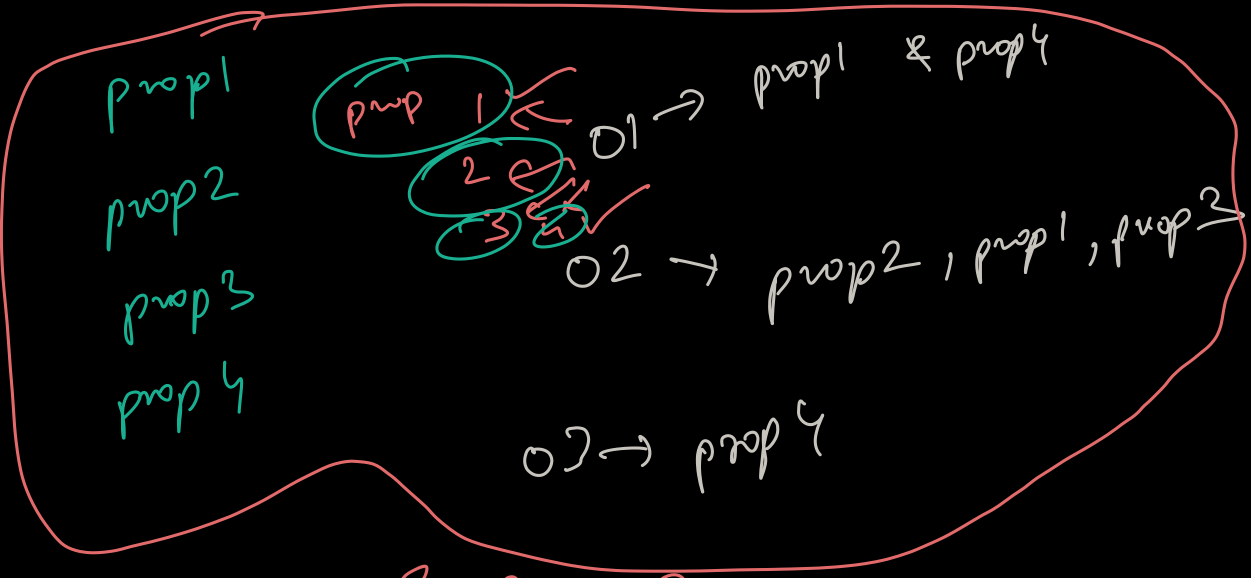
```
discard ← 1   1  1  1  1  1
  45 =        0  0  1  0  1  1  0  1
- 12 =        1  1  1  1  0  1  0  0
  ---------------------------------
  33          0  0  1  0  0  0  0  1
              7  6  5  4  3  2  1  0
```

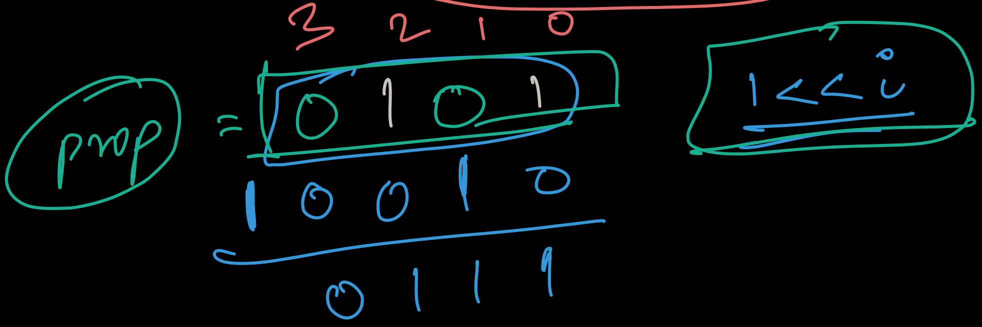
$2^5 + 2^0$

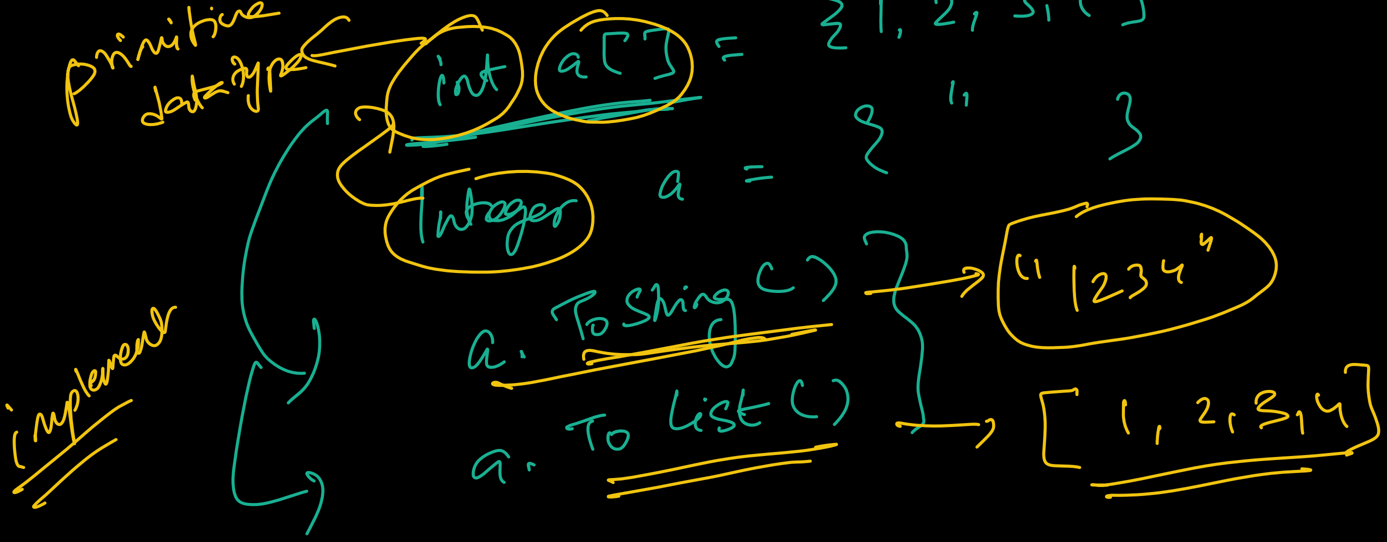
$= \boxed{33}$

flags:

```
enum fruit {
    apple = 0
    orange = 1
    banana = 2
}
```







Integer a

↓

int a