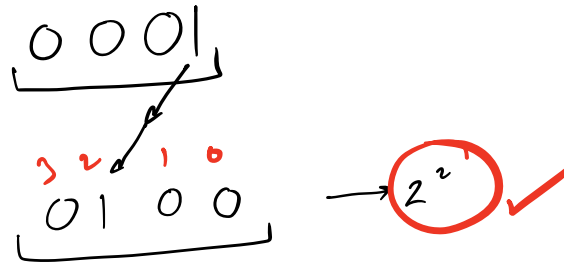


$$2^2 \longrightarrow 1 \ll 2$$



$$3^2 \xrightarrow{\text{X}} 1 \ll 3 \text{ X}$$

$$2^3 \xrightarrow{\checkmark} 1 \ll 3 \checkmark$$

$$N = 45 : 101101$$

OR

	5	4	3	2	1	0	
<u>OR</u>	1	0	1	1	0	1	: N
(1 < 2)	0	0	0	1	0	0	
	1	0	1	1	0	1	: N

OR

	5	4	3	2	1	0	
<u>OR</u>	1	0	1	1	0	1	: N
(1 < 4)	0	1	0	0	0	0	
	1	1	1	1	0	1	: <u>N + 2^4</u>

$N \rightarrow i^{\text{th}}$ bit in N was set
 $N \mid (1 \ll i) \rightarrow$ i^{th} bit in N was unset

Q → Given N, set its ith bit!

$$N = N | (1 \ll i)$$

XOR

Diagram illustrating the XOR operation to toggle the 3rd bit of N. N is 101101 (bits 5 to 0). Mask (1<<3) is 000100. The result N^(1<<3) is 101001, where the 3rd bit has been toggled from 1 to 0.

$N \wedge (1 \ll i) \rightarrow$ Flip/Toggle the ith bit!

AND

Diagram illustrating the AND operation to check if the 3rd bit of N is set. N is 101101 (bits 5 to 0). Mask (1<<3) is 000100. The result N & (1<<3) is 000100, where the 3rd bit is 1, indicating it is set.

$N \& (1 \ll i) \rightarrow$ $(1 \ll i)$: ith bit is set in N
 0 : ith bit is unset in N

Q Given N , unset its i^{th} bit if it's set!

if $((N \& (1 \ll i)) > 0) \{ \rightarrow i^{\text{th}} \text{ bit is set!}$

$$N = N \& (1 \ll i);$$

}



	5	4	3	2	1	0
\times	1	0	1	1	0	1
	1	1	0	1	1	1
<hr/>						
	1	0	0	1	0	1

$\rightarrow \sim(1 \ll i)$

$$N = N \& (\sim(1 \ll i))$$

Q Given N , check if the i^{th} bit is set or not?

	5	4	3	2	1	0
\times	1	0	1	1	0	1
	0	0	1	0	0	0
<hr/>						
	0	0	1	0	0	0

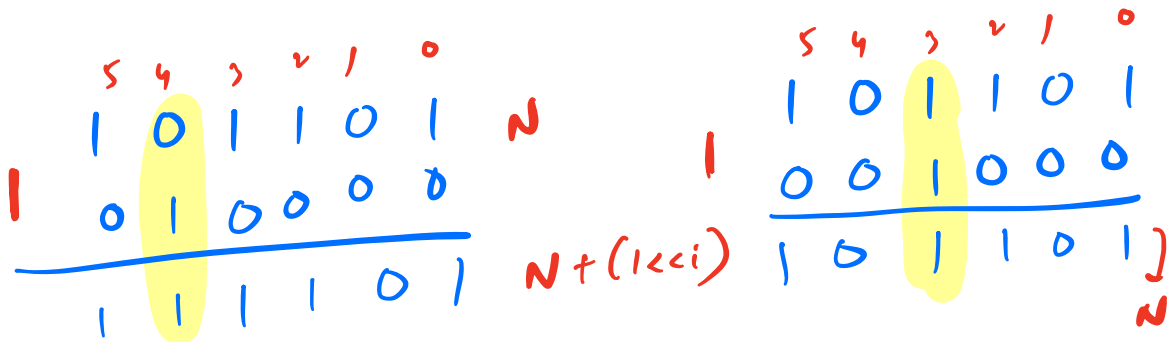
$$(N \& (1 \ll i)) = (1 \ll i)$$

	5	4	3	2	1	0
\times	1	0	1	1	0	1
	0	1	0	0	0	0
<hr/>						
	0	0	0	0	0	0

$$(N \& (1 \ll i)) = 0$$

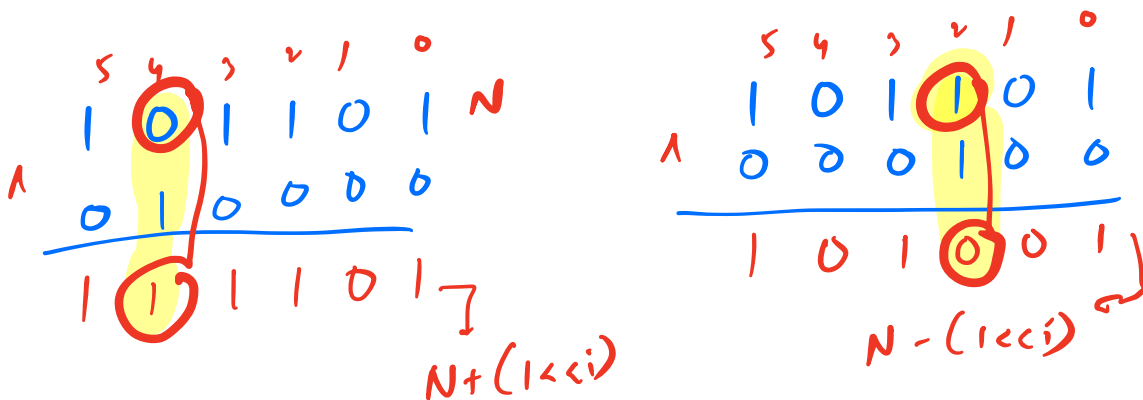
if $(N \& (1 \ll i)) > 0$
 \rightarrow i th bit is SET

else \rightarrow i th bit is UNSET

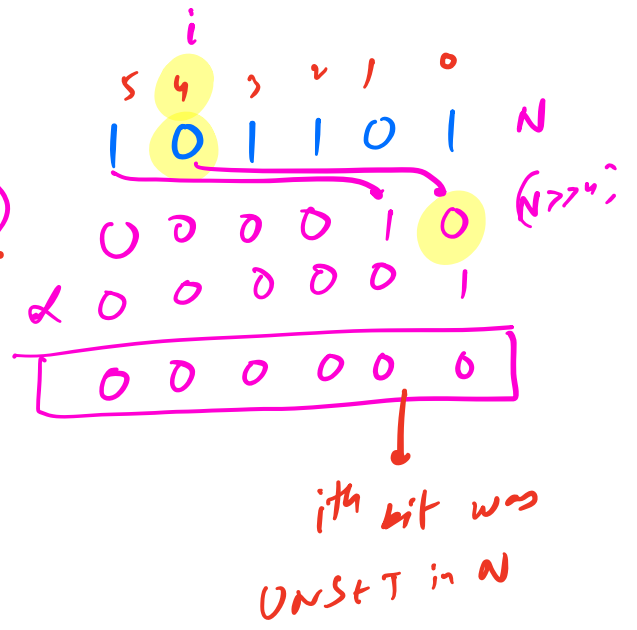
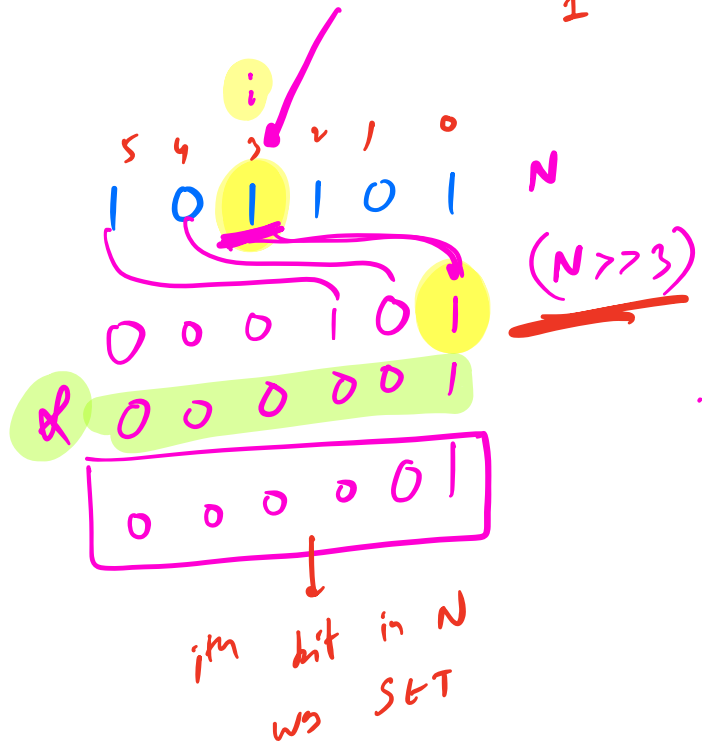
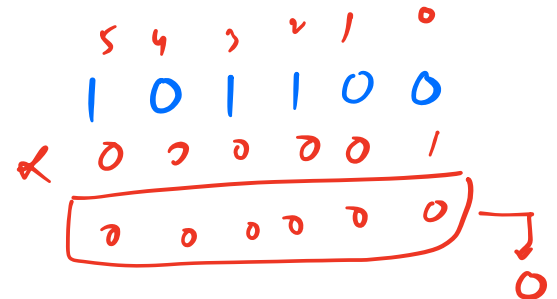
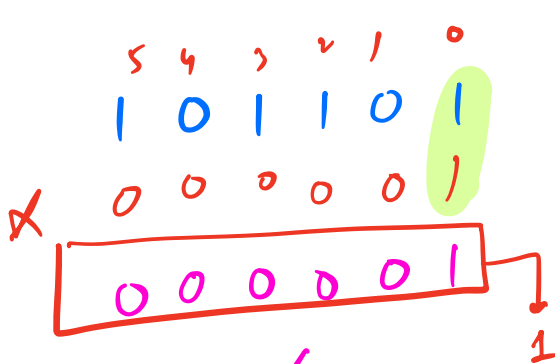


if $((N | (1 \ll i)) > N)$
 \rightarrow i th bit is UNSET

else \rightarrow i th bit is SET



if $(N \& (1 \ll i)) < N$
 \rightarrow i th bit is SET
 else \rightarrow NOT SET



if $((N \gg i) \& 1) > 0$
 \hookrightarrow ith bit in N was SET

else \hookrightarrow UNSET

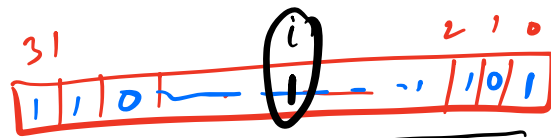
Q Given a number N . Count the total no. of set bits!

$$N = (6)_{10} \longrightarrow (110)_2 : 2 //$$

$$N = (7)_{10} \longrightarrow (111)_2 : 3 //$$

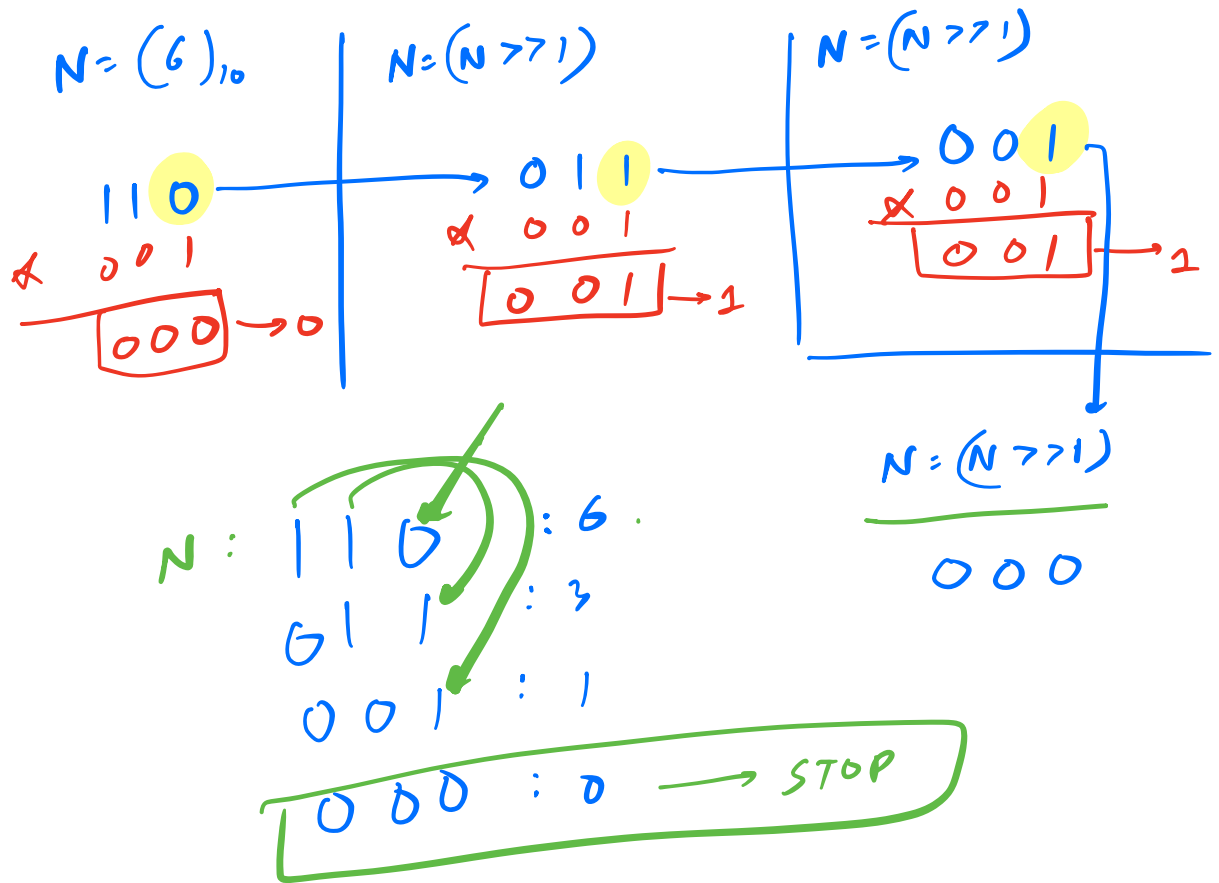
$$N = (12)_{10} \longrightarrow (1100)_2 : 2 //$$

$N: \text{int} \rightarrow 32 \text{ b}$



$\text{cnt} = 0$
 $\text{for } (i = 0; i < 32; i++) \{$
 $\quad \text{if } ((N \& (1 << i)) > 0) \{$
 $\quad \quad \text{cnt}++;$
 $\quad \}$
 $\}$
 ret cnt;

$\nearrow \text{int} \quad | \quad 64: \text{long}$



```

cnt = 0;
while (N > 0) {
    if ((N & 1) > 0) {
        cnt++;
    }
    N = (N >> 1);
}

```

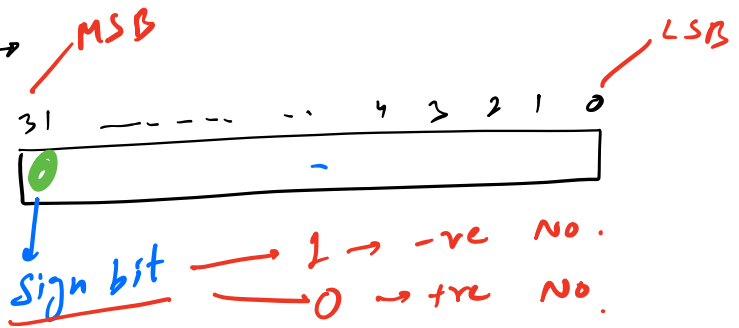
Time Complexity: $O(\log N)$

Space Complexity: $O(1)$



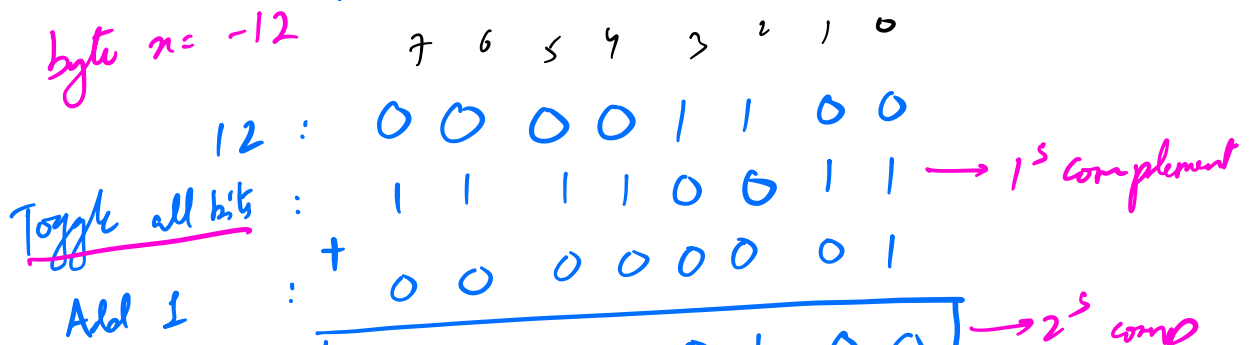
Negative Numbers

int $x = -12$



8 bit No

byte $x = -12$



$$\begin{aligned}
 & -2^7 + 2^6 + 2^5 + 2^4 + 2^2 \\
 & -128 + 64 + 32 + 16 + 4 = -12
 \end{aligned}$$

$$0 \times -2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + \dots$$

$$\begin{array}{r}
 N = 0 : \quad 00000000 \\
 \quad \quad \quad 11111111 \\
 + \quad \quad \quad 00000001 \\
 \hline
 -0 \rightarrow \boxed{00000000}
 \end{array}$$

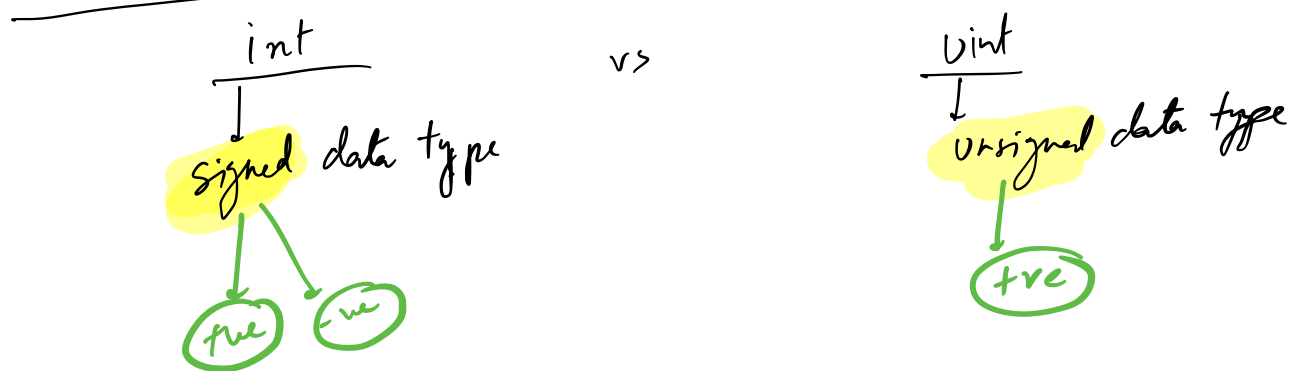
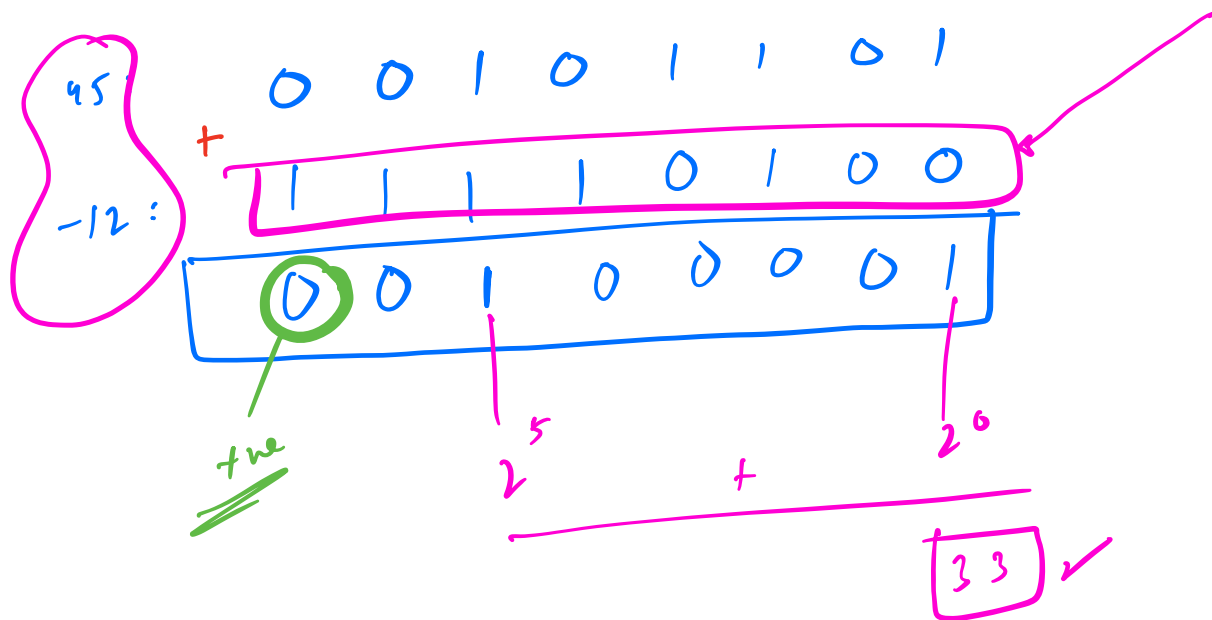
$$\begin{array}{r}
 N = 45 \quad 00101101 \\
 \quad \quad \quad 11010010 \\
 + \quad \quad \quad 00000001 \\
 \hline
 \boxed{11010011} \rightarrow -45 \\
 + \quad \boxed{00101101} \rightarrow 45 \\
 \hline
 \boxed{00000000} \rightarrow 0
 \end{array}$$

I

Subtraction of binary Numbers →

$$45 - 12$$

$$45 + (-12)$$



⑧ MAX no. an int can store?

31 — — — — — 2 1 0

0 1 1 1 1 1 1 1 1 1

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

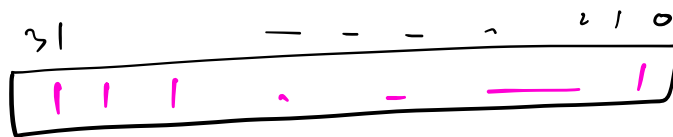
$2^{31} - 1 \sim 2 \times 10^9$

long $\rightarrow 2^{63} - 1 \sim 9 \times 10^{18}$

④



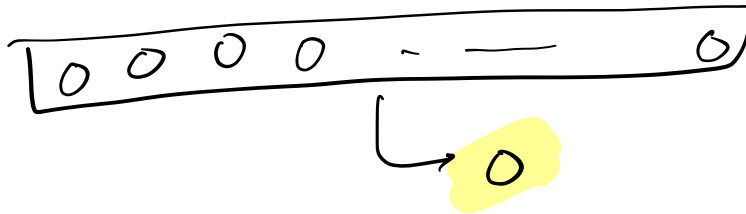
long $\rightarrow -2^{63} \sim -9 \times 10^{18}$



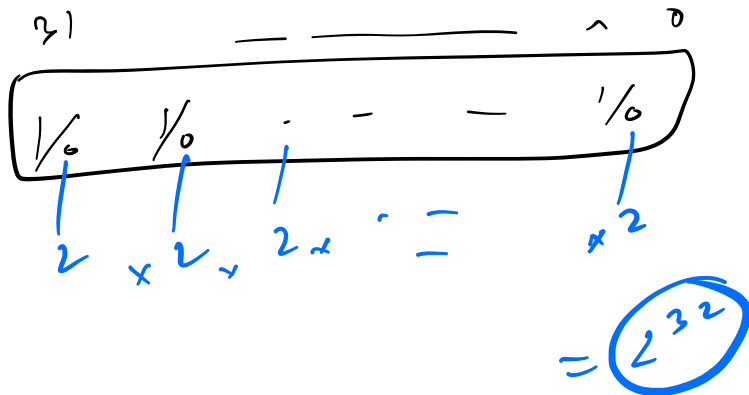
$$2^{31} + 2^{30} + 2^{27} + \text{---} + 2^0$$

$\hookrightarrow 2^{3^2} - 1$

$\sim 4 \times 10^9$



uint $[0, 2^{32}-1]$ → 2^{32}
 int $[-2^{31}, 2^{31}-1]$ → 2^{32}



Q Given an array. Find the sum of the Array!

long 10^{18}
 int sum = 0;
 for (i=0; i < N; i++) {
 sum += A[i];
 }
 ret sum;

CONSTRAINTS →

$1 \leq N \leq 10^5$

$1 \leq A[i] \leq 10^9$

$10^9 \times 10^5 = 10^{14}$

$10^9 \times 10^5$

10^{14}

$$\begin{array}{r} 0001 \\ + 0001 \\ \hline 0010 \end{array} \rightarrow 2$$

$$\begin{array}{r} 0001 \\ + 0001 \\ + 0001 \\ \hline 0011 \end{array} \rightarrow 3$$

$$\begin{array}{r} 0001 \\ + 0001 \\ + 0001 \\ + 0001 \\ \hline 0100 \end{array} \rightarrow 4$$

$$\begin{array}{r} 0001 \\ + 0001 \\ + 0001 \\ + 0001 \\ + 0001 \\ \hline 0110 \end{array} \rightarrow 6$$

$$\begin{array}{r} 0001 \\ + 0001 \\ + 0001 \\ + 0001 \\ + 0001 \\ + 0001 \\ \hline 1000 \end{array} \rightarrow 8$$

 overflow
 -8

Q Given 2 `ints a & b`, ret `a * b`!

X `int ans = a * b;`
`ret ans;`

$\sim 10^9 \times \sim 10^9 \rightarrow \sim 10^{18}$

X `long ans = a * b;`
`ret ans;`

`a * b`
`int & int`
overflow!
`int`

implicit type cast
`32 int + long`
`64 long`
type promotion long

`long ans = long(a * b)` overflow!

`long ans = long(a) * b` Explicit type cast

or `long(a) * long(b)`

`a * 1L * b`

