Bitwise Operators

1. AND &
2. OR |
3. XOR ^
4. NOT ~ (unary op)
5. Left shift <<
6. Right Shift >>

&& logical and
|| logical OR
! Logical Not

boolean expression

5 > 3 && 7 > 4
T T
T

```
101
111
---
101
```

5 & 7 => 5

operate on bits

Q1. Check / Get the ith bit of number.

$n = 13$

get last bit ⇒ $n \& 1$

```
Get I^th Bit (n, i) {

return (n >> i) & 1

}
```

$n = 13$

$= 1\ 1\ 0\ 1$ (bit positions 3 2 1 0)

$i = 2$

$0\ 0\ 0\ 1\ 1\ \otimes\ *$

↑ last_bit

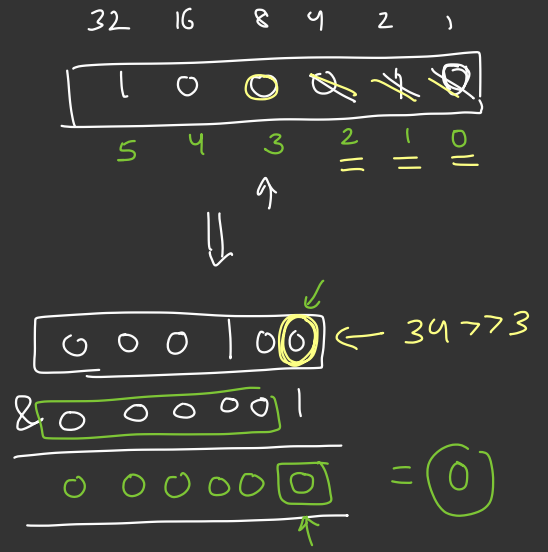
$n = 34$

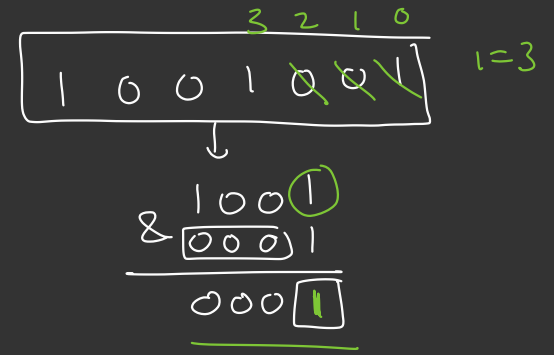
$i = 3$

$(34 >> 3) \& 1$

$i^{th}\ bit = (n >> i) \& 1$

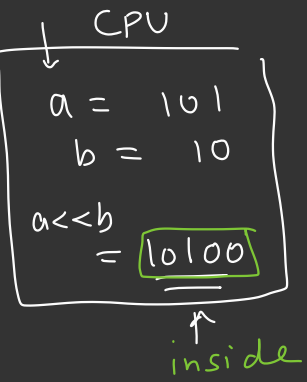
$n >> 3$

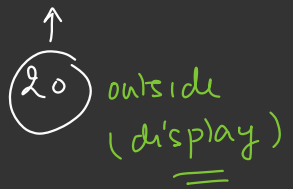




```
int  a = 5
int  b = 2

print( a <<b )
```



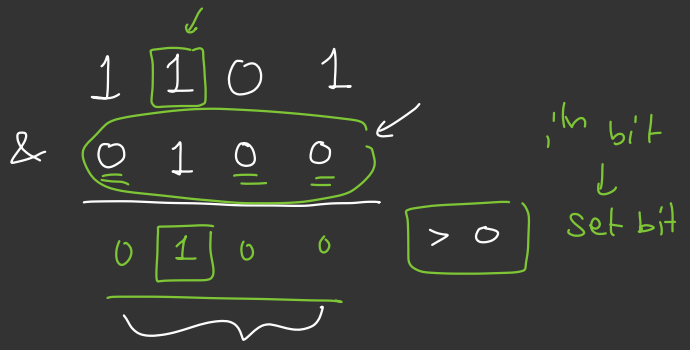


```
GetIthBit(n, i){
    return (n>>i) & 1
}
```

always produces 0 or 1

$n = 13$

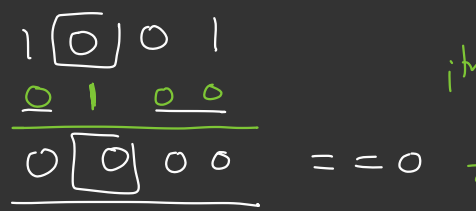
$i = 2$



$(n \,\&\, (1<<i)) == 0$ 0 bit

> 0 1 bit

$n =$



```
getIthBit(n, i){
    if((n & (1<<i)) > 0)
        return 1
    else
        return 0
}
```

O(1)

n =

i = 5

n = 1 0 0 0 1 0 0 (5 4 3 2 1 0)

&

1<<i = 0 1 0 0 0 0

⇒ 1 0 0 0 0 0

return n & (1<<i)

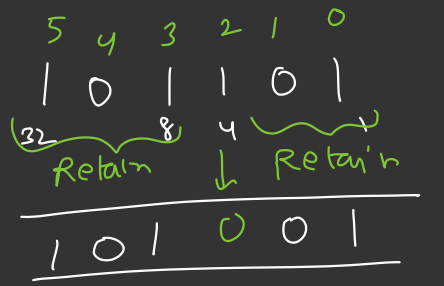
NO

```
 1 1 0 1
 0 1 0 0
 -------
 0 1 0 0   → 4
```

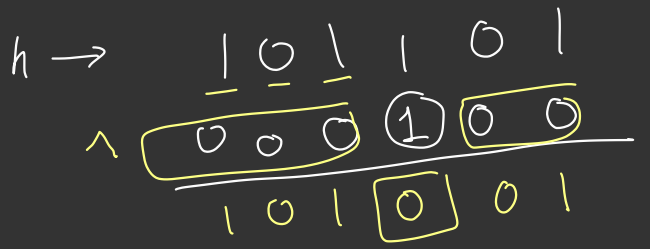
(Q) Unset i^{th} bit of a number.

→ if bit is 0 → No change
→ " is 1 → 0

N = 45



i = 2



1<<i

Set Bit

```
if (checkbit(n,i)==1){
    n = n ^ (1<<i)
}
```

only when i^{th} bit is 1

Q Count total no of set bits (bit == 1) [one] in number N

N = 45 → 0 0 0 1 0 1 1 0 1

1 + 0 + 1 + 1 + 0 + 1 = 4

ans = 0

ans = 0 + 1 + 0 + 1 + 1 + 0 + 1 = 4

```
while (N > 0) {
    last_bit = (N & 1)
    ans = ans + last_bit
    n = n >> 1
}
print(ans)
```

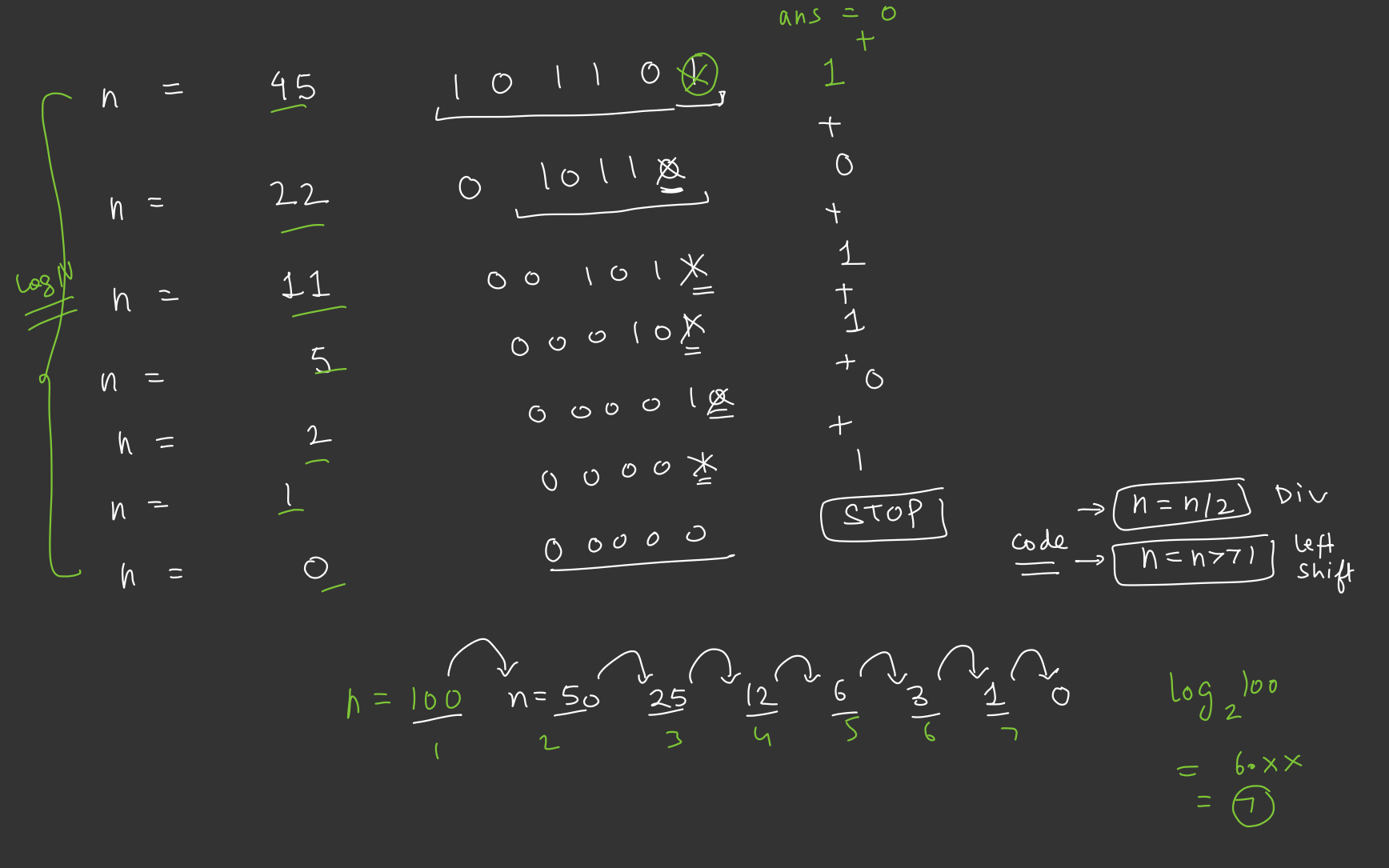
last_bit → integer 0 or 1

equivalent n = n/2

1 0 1 0 1 0 1

Time Complexity O(Log N)

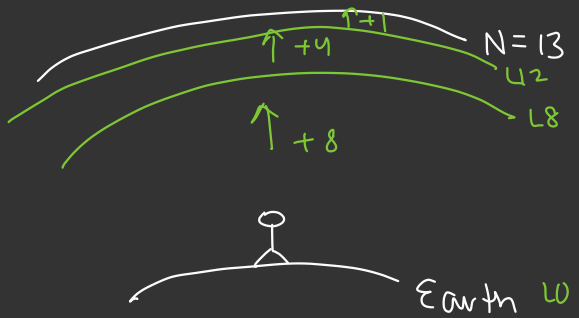
≤ 32


n = 45
n = 22
n = 11
n = 5
n = 2
n = 1
n = 0
Log N
1 0 1 1 0 ✗
0 1 0 1 1 ✗
0 0 1 0 1 ✗
0 0 0 1 0 ✗
0 0 0 0 1 ✗
0 0 0 0 0
ans = 0
1
+ 0
+ 1
+ 1
+ 0
+ 1
STOP
code
n = n/2 Div
n = n>>1 left shift
n = 100 50 25 12 6 3 1 0
1 2 3 4 5 6 7
log2 100
= 6.xx
= 7

Q. There is a battle going a N levels above the surface earth. The hero which in at earth can take jumps in power of 2. Minimum jumps he will need to reach level N.

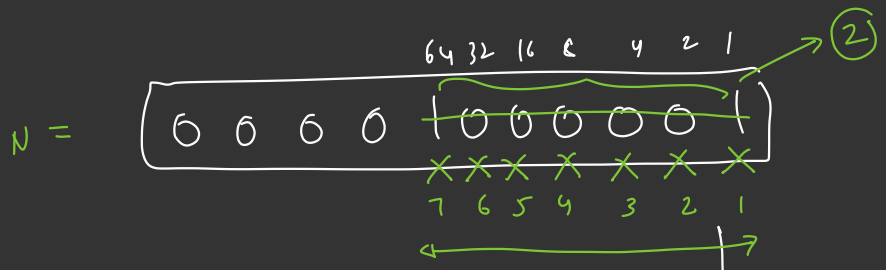
jump sizes = [1, 2, 4, 8, 16, ...]

3 jumps



13 = (1)(1)0(1)
8 + 4 + 1

Set Bits → Set count bit = (3)



How many its itr

7 iterations.

Counting set bits takes log N iterations

No of sets bits ≠ log N

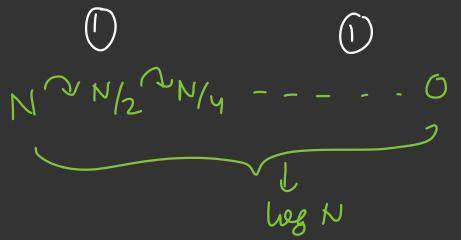
count

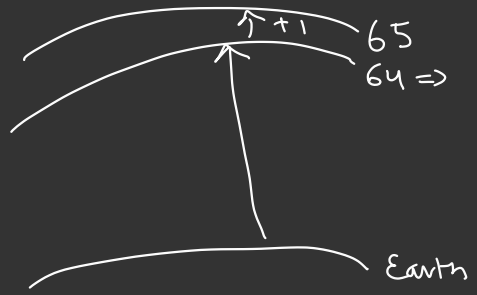
Set Bits = 2

2 jumps

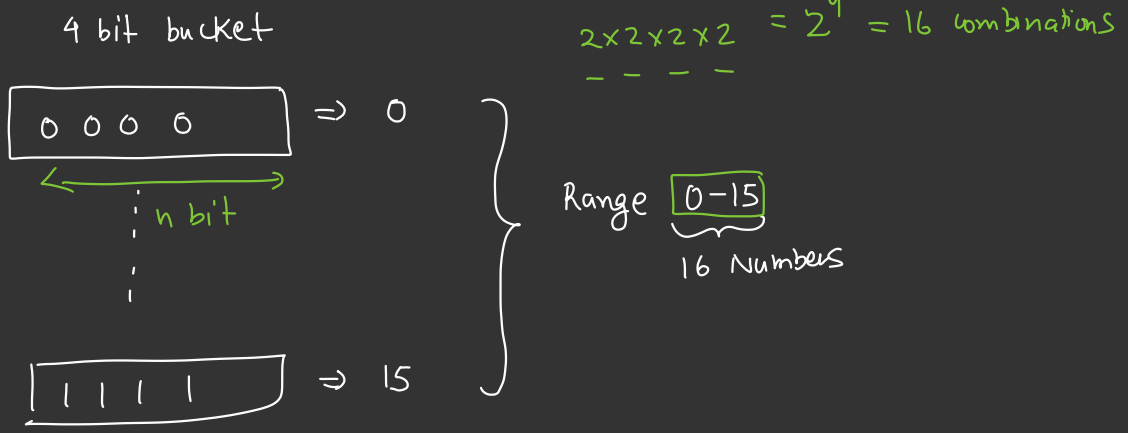
Log 65 = 6.xx

= 7

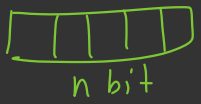




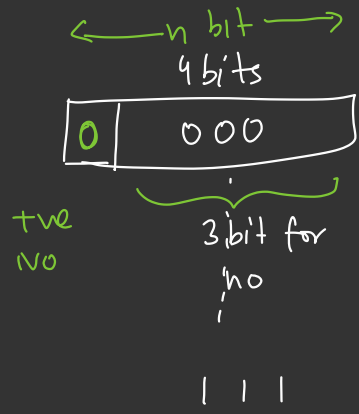
Negative Numbers



Concept of Sign bit was introduced



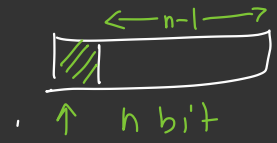
$0 \longleftrightarrow 2^n - 1$



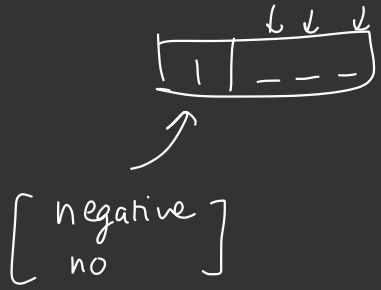
(0 → 7) 8 combinations

$2^{4-1} - 1$

$= 7$



0 to $2^{n-1} - 1$

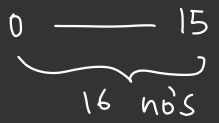


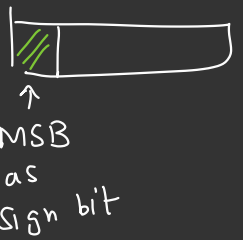
8 combinations

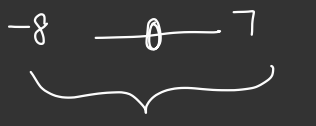
Range

(-8, -7, ... -1)


4 bits


0
15
16 nos


MSB
as
Sign bit

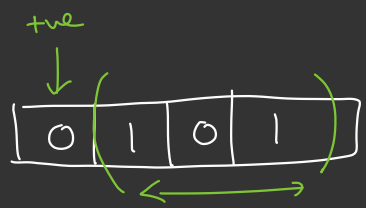

-8
0
7
16 Numbers

How -ve no's are actually stored ?

↳ 2's compliment form

↓ if we add negative no's in bin's from result of subtraction.

(5)



-(5)

1 0 1 0

\+ 1

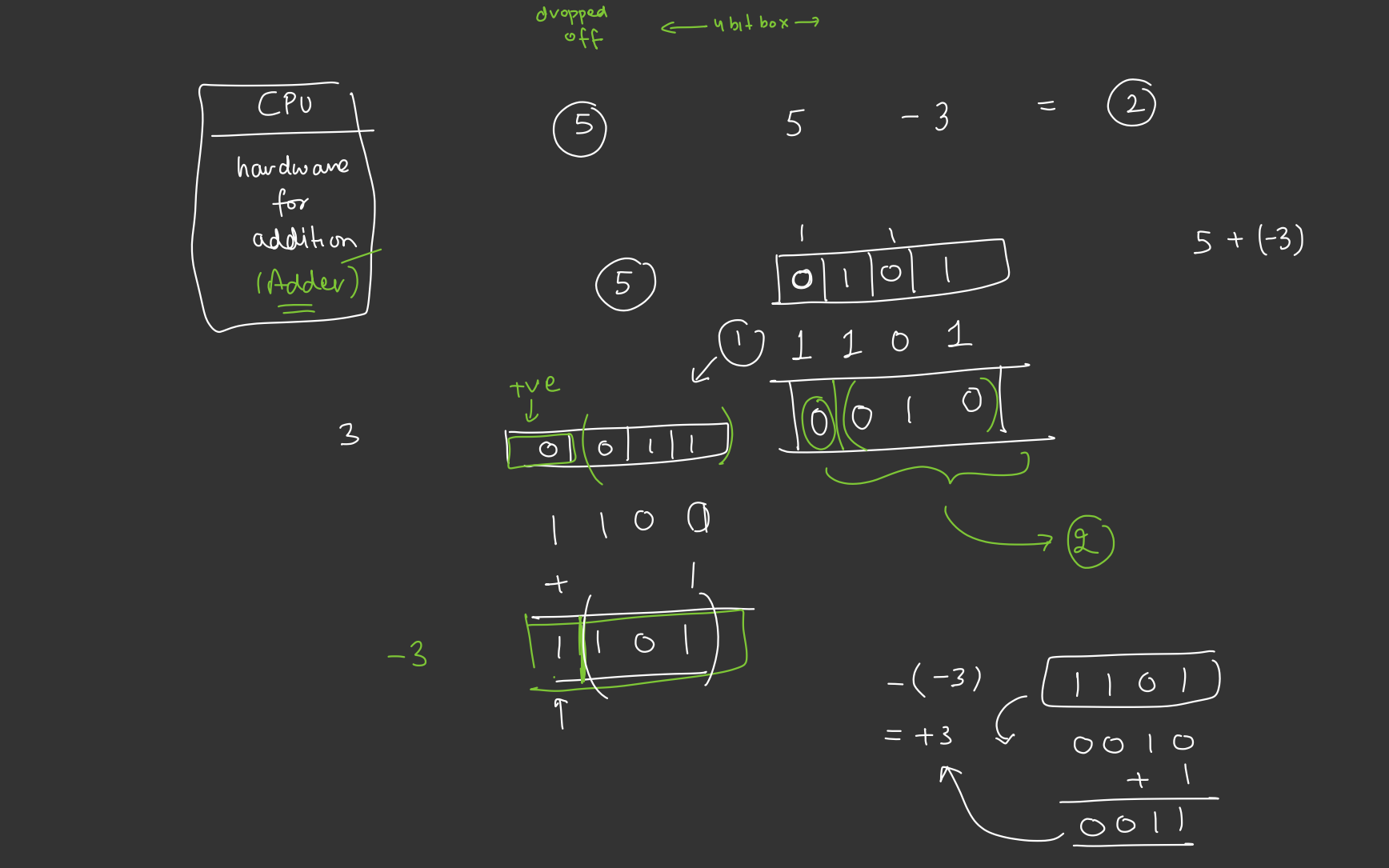
1 0 1 1

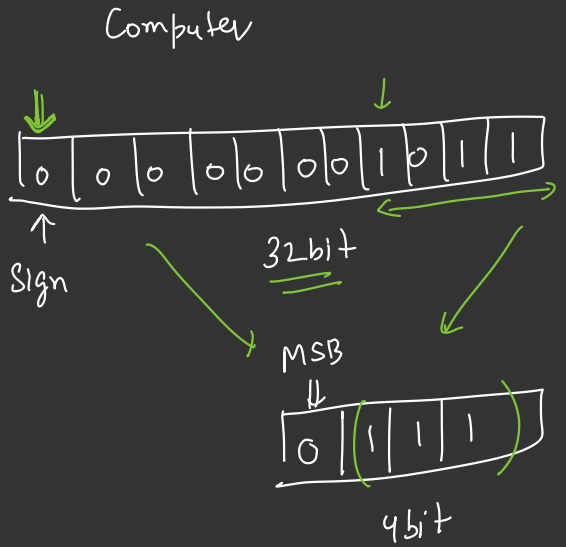
① Flip all bits of 5
② Add 1

5 - 5 ~~subtraction~~

5 + (-5)

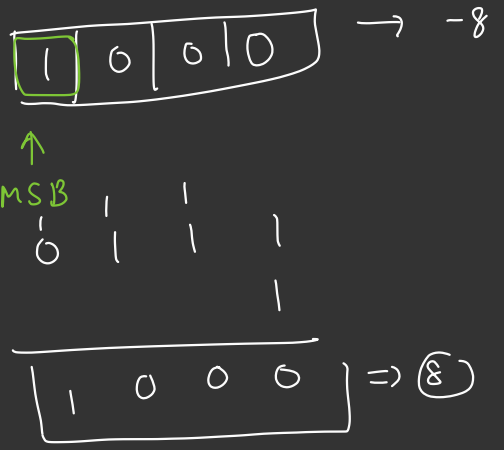
= 0


dropped off
4 bit box
CPU
hardware for addition (Adder)
5
5 - 3 = 2
5 + (-3)
5
0 1 0 1
1 1 0 1
0 0 1 0
+ve
0 0 1 1
3
1 1 0 0
+ 1
1 1 0 1
-3
-(-3)
= +3
1 1 0 1
0 0 1 0
+ 1
0 0 1 1


Computer
0 0 0 0 0 0 0 0 1 0 1 1
Sign
32bit
MSB
0 1 1 1
4bit

Read ⇒

his NO



$-(-x)$

$\Rightarrow +x$

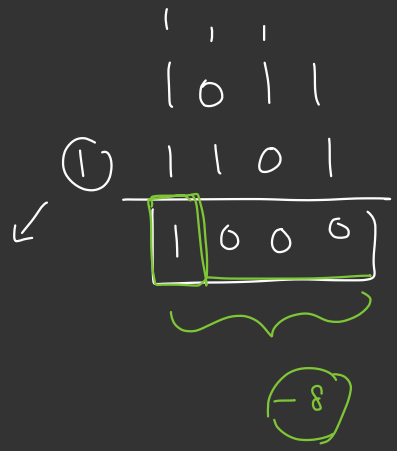
$-5 + (-3)$

$5 \quad 0101$

$-5 \quad 1010$

$\quad\quad +1$

$\quad\quad 1011$



Java

int x = 44

y = -76

(Standard)

32nd bit Reserved

is always MSB

=) 1 -ve

0 +ve

getIthBit(N, i)

Yes ~~No~~

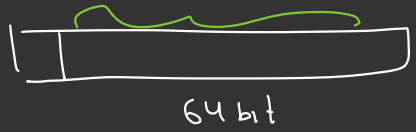
Range of No's

int

$-2^{31} \longleftarrow 0 \longrightarrow 2^{31}-1$

-2×10^{9} $\quad \approx 2 \times 10^{9}$

long



$-2^{63} \longleftarrow 0 \longrightarrow 2^{63}-1$

$-2 \times 10^{18} \longleftrightarrow \approx 2 \times 10^{18}$

Q Calc sum of all elements of an array of size N

Constraints Analyze

$A[i] \leq 10^6$, $N = 10^5$

```
long ans = 0
for( int x : arr){
    ans = ans + x
}
```

$[\underbrace{10^6 + 10^6 + \cdots}_{10^5}$

$\leq 10^{11}$

(A) given two integers, find $a * b$

$a \leq 10^9$
$b \leq 10^9$

✗ (1) `print( a * b )` — int int → int — ✗ overflow

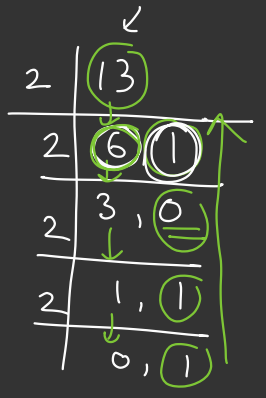
System assumes output is an int

✗ (2)
```
long ans = a * b
print(ans)
```
a, b: int int → int (??) ↝ 10^{18}

~~(3)~~
```
long ans = long(a) * b
print(ans)
```
long * int → long

Counting Set Bits

13 = 1 1 0 1



faster

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

last bit

```
while (N > 0){
    rem = N%2
    ans = ans + rem
    N = N/2
}
```

5 bits

5 | 0 0 1 0 1

-5 | 1 1 0 1 1

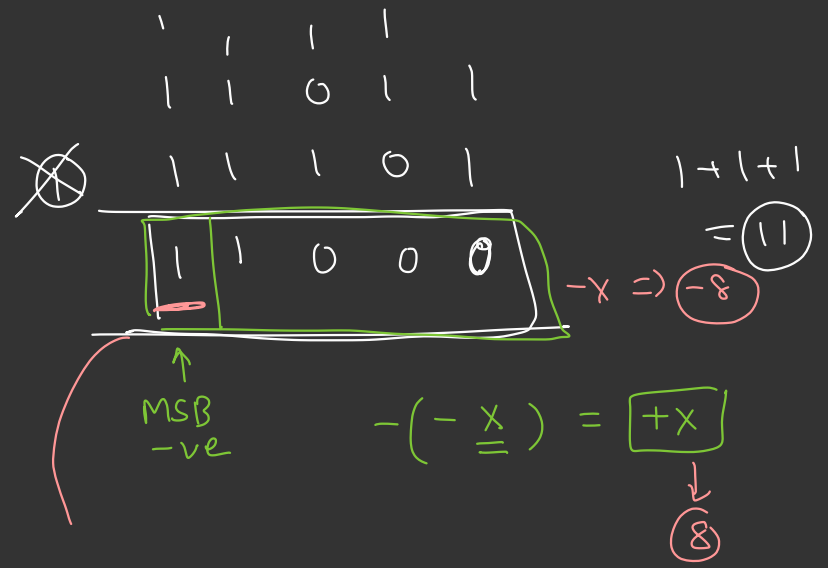
(-5) * (-3)

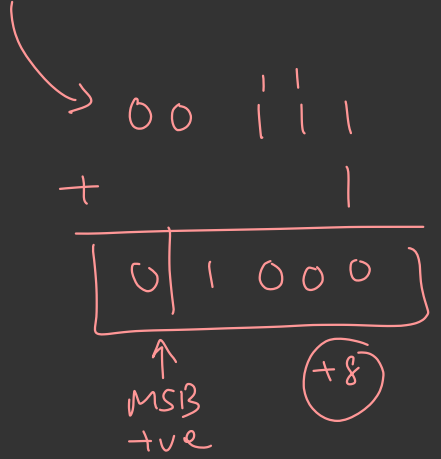
3 | 0 0 0 1 1

-3 | 1 1 1 0 1

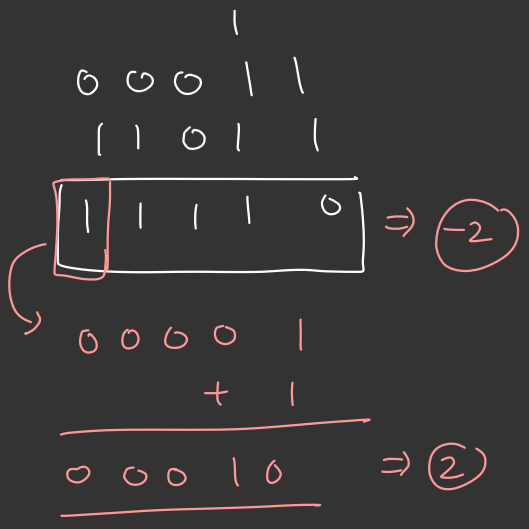
(-5)
+(-3)

-8





$3 + (-5)$



Decimal to Binary

N = 13

output =

1000
100
00
1

p=1 1101

2	13
2	6, 1
2	3, 0
2	1, 1
	0, 1

1x1 = 1

0x10 = 00

1x100 = 100

1x1000 = 1000

N=13 ans = 0, p = 1

while (N>0){

rem = N%2 → N&1

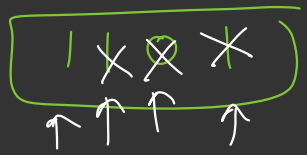
ans = ans + p*rem

p = p*10

N = N/2 → N=N>>1

}

13



$$
\begin{array}{r}
1\times1 \\
+\ 0\times10 \\
+\ 1\times100 \\
+1\times1000 \\
\hline
1101
\end{array}
$$

```
while(N>0){
    bit = N&1
    ans = ans + p*bit
    p = p*10
    N = N>>1
}
```