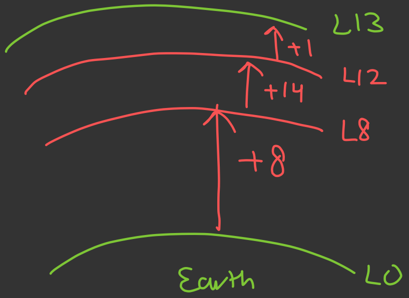
Bitmasking

warm up

There is a enemy at N levels above the surface of the earth. You are a superhero, standing a earth (level 0) and you can take jumps in powers of 2. Min jumps that would be needed to reach the enemy.

1,2,4,8,16,32---

N = 13



$$13 = \underset{2^3}{1}\ \underset{2^2}{1}\ 0\ \underset{2^0}{1}$$

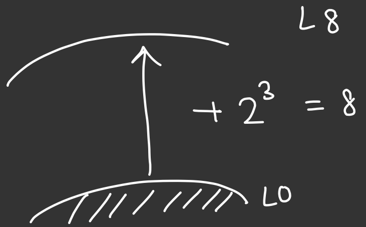
$$= 8 + 4 + 1$$

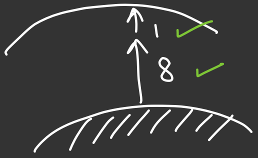
$$= \boxed{13}$$ → Always be the minimum

Logic

↓

Count the no of set bits in $\boxed{N}$ (Binary Rep)





$$9 = \boxed{\overset{2^3}{1}\;0\;0\;\overset{2^0}{1}}$$

$$8 + 1 = \textcircled{9}$$

Set bits in

bin rep in N

$N = 13$

$$\begin{array}{r} 1\;1\;0\;\cancel{1} \\ \& \qquad\quad 1 \\ \hline 1 \\ \hline \end{array}$$

$$\begin{array}{r} 1\;1\;\cancel{0} \\ \&\quad 1 \\ \hline +\quad 0 \\ \hline \end{array} \quad \begin{array}{r} 1\;\cancel{1} \\ \&\; 1 \\ \hline +\; 1 \\ \hline \end{array} \quad \begin{array}{r} 1 \\ \& \; 1 \\ \hline +\; 1 \\ \hline \end{array}$$

$= \textcircled{3}$ jumps

N&1 → gives me the value of last bit in N

$N = \boxed{6}$

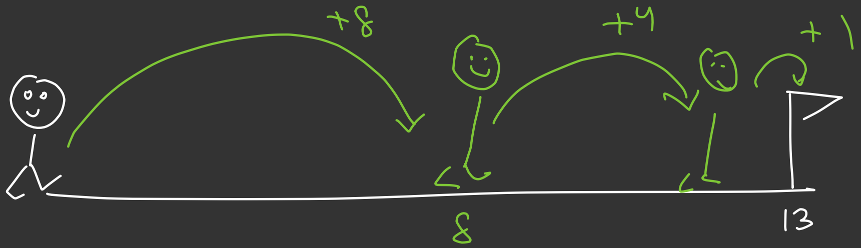
$6\&1 \to 0$

$$\begin{array}{r} 0000000\,1\,1\,\textcircled{0} \\ \& \; \underbrace{00000000}\;1 \\ \hline 0000000\;\boxed{0} \\ \hline \end{array} \; \textcircled{0}$$

$N = 9$

N&1

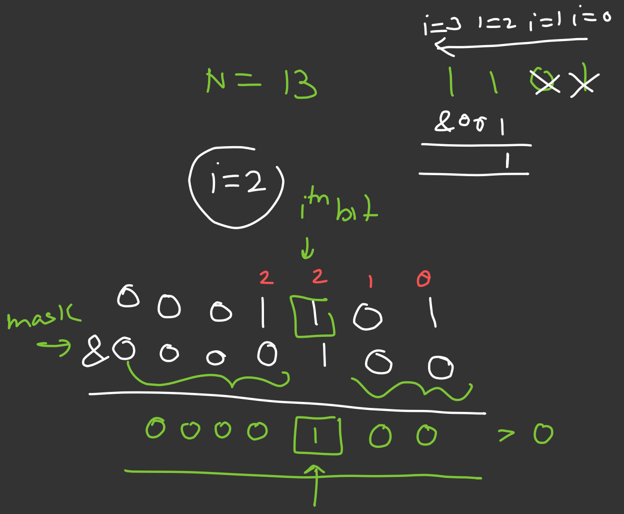
$$\begin{array}{r} 0000100\,\textcircled{1} \\ \& \; \underbrace{0000000}\;1 \\ \hline 0000000\;\boxed{1} \\ \hline \end{array} \; \textcircled{1}$$



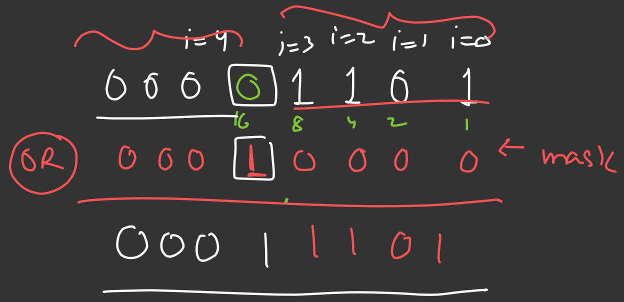
Common Operations involving Bitwise Operators

```
int getIthbit(no, i) {
        return (No>>i)&1 ;
}
```

```
int getIthbit(no, i) {
        mask = 1<<i

        if (no & mask >0) {
                return 1,
        }
        else {
                return 0,
        }
}
```



```
Set I^th Bit ( N , i ) {

    mask = 1<<i

    return N | mask

}
```

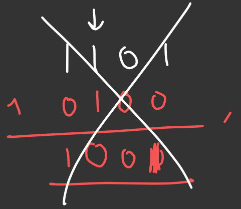


& → clearing bit & 0

| → setting bits | 1

$13+16 \Rightarrow 29$

$N + (1<<i)$



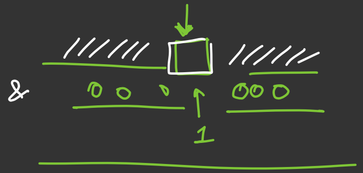
Toggle Bit

↓

1 → 0

0 → 1

} XOR with mask



$0 \wedge 1 = 1$

$1 \wedge 1 = 0$

```
Toggle I^th Bit( N, i ){

    mask = 1<<i ;

    return N ^ mask

}
```

TODO:

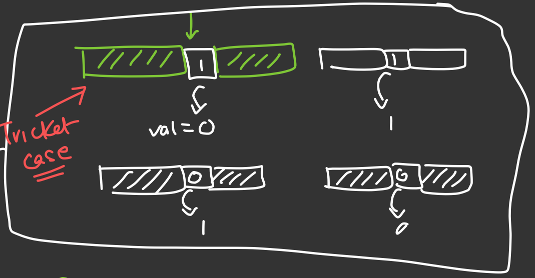
```
Update I^th Bit ( N, i, v ) {
    step1 [  mask = ~(1<<i);  ]
             N = N & mask;

    step-2   N = N | (v<<i)  ] v==0 → No change

             return N;
}
```

1 ↑ ↗ 0

1<<i

10000 → 0 0000 →

v → 0 or 1



Tricket case

1<<i
if N==1

① 3 2 1 0

1 1 0 1 → 1001 = ⑨

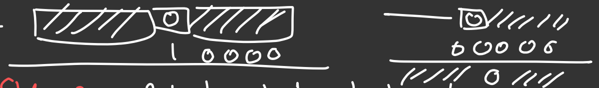
1 1 0 1 → 1111 ⇒ 15

update (13, 2, 0)

update (13, 1, 1)

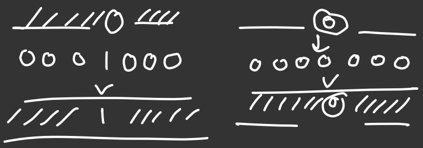
Step-1 make ith bit as 0 (Clear ith Bit)

```
clear BitI( N, i ){
    ___
    ___
    ___
    ___
}
```

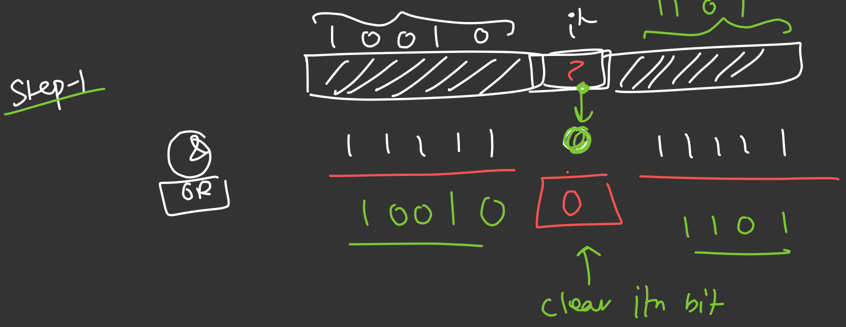
N –


Step-2 Set the val at loc i.

N | v<<i



Step-1



clear ith bit

$mask = \sim(1 << i)$

$N = N \,\&\, mask;$

Challenge

Given a No, clear last I bits ?

$11\overrightarrow{\cancel{01}}^{\,ibits} \Longrightarrow \overleftarrow{1100}$

N = 13

i = 2

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

↓

1100

12

great

$(N >> i) << i$

Another way

$$\begin{array}{rl} & \overbrace{1\,1\,0\,1\,1\,0}^{Retain}\ \overleftarrow{1\,1\,1\,0}^{\,i} \\ \& & 1\,1\,1\,1\,1\,1\ \underbrace{0\,0\,0\,0}_{gone} \\ \hline \rightarrow & 1\,1\,0\,1\,1\,0\ 0\,0\,0\,0 \end{array}$$

$mask = (-1 << i)$

$N = N \,\&\, mask$

$-1 = \boxed{1}1111111111$ $\quad$ (-1)

2s Compliment

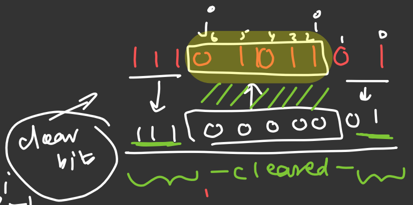
$-1 << i =$ ~~111~~ $1111111\;\underbrace{0000}_{i}$

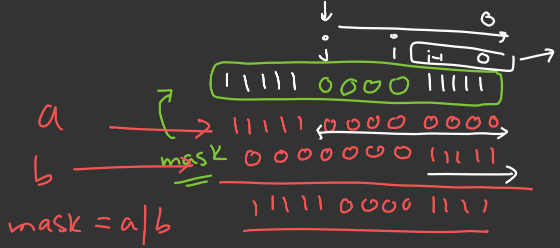
Flip all bits + 1

$\Rightarrow$ 000000000
+ 1

000000001

```
clear Bits In Range (N, i, j){
    a = -1 << (j+1) ,
    b = (1<<i) -1 ;        // 2^i -1
    mask = a|b;
    n = n & mask;
    return n,
}
```





$1 << i \Rightarrow 2^i$

1 → 01

2 → 10

3 → 11

4 → 100

5 → 101

6 → 110

7 → 111

8 → 1000

⋮

15 → 1111

16 → 10000

⋮

$31 = \overset{16+8+4+2+1}{\boxed{1\ 1\ 1\ 1\ 1}}$

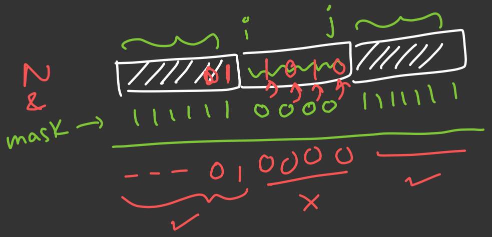
$2^2-1 = 3 = \boxed{11}$

$2^3-1 = \boxed{111}$

→ $\boxed{000000\ \overleftrightarrow{1111}}$

$2^4-1 = \boxed{1111}$

2^5-1



&1 → Retain Bit

&0 → Clear Bit

$2^4-1 \Rightarrow 16-1 \Rightarrow \boxed{15}$

1<<4 − 1

= 10000 − 1

= 01111

$2^K-1 \Rightarrow \boxed{0000\ \overset{K}{1111}}$

Q

Given 2N + 2 Numbers where every no is coming twice except 2 unique NO find out the unique NO's (Bitwise)

15 Min Break

2, 5, 7, 7, $\boxed{9}$, 5, $\boxed{3}$, 2

once

$\boxed{9,3}$

Welcome back

$XOR = \cancel{2}\wedge\cancel{5}\wedge\cancel{7}\wedge\cancel{7}\wedge 9\wedge\cancel{5}\wedge 3\wedge\cancel{2}$

↓

res = $9 \wedge 3$ // $\boxed{10}$

↓

will altleast one set bit → find out the index of ~~the~~ any set bit

$$\begin{array}{r} 1001 \\ 0011 \\ \hline 1010 \\ 8\,4\,2\,1 \end{array}$$

= $\boxed{10}$

Tricky

i=3	i=2	i=1	i=0
1	0	1	0

idx

$$\begin{array}{rr} a & 1101 \\ b \;\wedge & 1011 \\ \hline & 0110 \end{array}$$

Any set Bit

must be present only in one of the no's

→ sorting

$2, 2, \boxed{3}, 5, 5, 7, 7, \boxed{9}$

N Log N

$$\overset{10}{2},\ \overset{101}{5},\ \overset{111}{7},\ \overset{111}{7},\ \overset{1001}{9},\ \overset{101}{5},\ \overset{11}{3},\ \overset{10}{2}$$

XOR

have set bit at idx $A \rightarrow \cancel{2}, \cancel{7}, \cancel{7}, \boxed{3}, \cancel{2}$ $2N+1$ $\boxed{3}$

don't have a set bit $B \rightarrow \cancel{5}\ 9\ \cancel{\boxed{9}}, 5$ $2N+1$ $\boxed{9}$

$A = 0$

$A \wedge \cancel{2} \wedge \cancel{7} \wedge \cancel{7} \wedge 3 \wedge \cancel{2}$

$0 \wedge 3 = \boxed{3}$

$ARR[N] = \{2, 5, 7, 7, 9, 5, 3, 2\}$

step-1

↓

O(N)

```
res = 0
for (i=0; i<N; i++) {
    res = res ^ arr[i];
```

$0 \wedge 2 \wedge 5 \wedge 7 \wedge 9 \wedge 5 \wedge 3$
$\wedge 2$

0—32

step-2

find out pos of set bit.

//res = 9^3 = 10

O(1)

```
temp = res,
pos = 0
while( (temp&1)==0 ){        max 32 times
    pos = pos+1
    temp = temp>>1;
```

3

res → // 10 → 1 0 1 ~~0~~

p=1 p=0

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

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

// pos → 1 → only 1 of the uniq have a set bit,

A B

Step-3

Based upon idx divide no's into 2 Sets

int A = 0 ^ ~~2~~ ^ ~~7~~ ^ 3 ^ ~~7~~ ^ ~~2~~ 3

int B = 0 ^ ~~5~~ ^ 9 ^ ~~5~~

O(N)

```
for ( i=0 ,   i< n ;   i++ ) {
        if ( getIthBit ( arr(i) , pos ) ==1) {
             → A = A ^ arr(i);
        }
}

B = res ^ A

print (A, B)
```

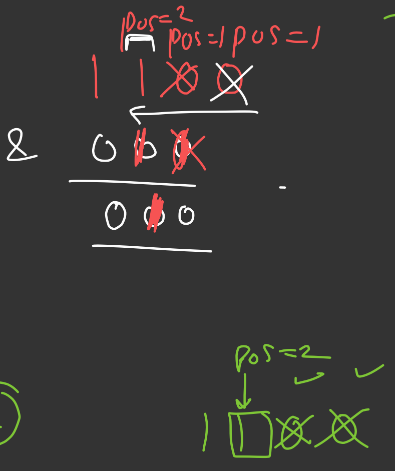
$res = \cancel{UN1} \wedge \boxed{UN2} \wedge \cancel{UN1}$

overall time → O(N)
Space → O(1)

Best soln

2nd

```
N = 12

pos = 0          N&1 → 1 stop

while ((N&1)==0){
    pos = pos+1
    N = N>>1

}

print(pos),          // 2
```



Day -1	Day - 2	Day - 3	Day - 4
1 unit	2	4	8

Break & make ✓

+ Practice

$A = \underline{14 \text{ units}}$ → 3 Days

Count set Bits

Power of 2

$8 + 4 + 2$

1	1	1	0
8	4	2	1
✓	✓	✓	

o Reverse bits of 32 bit ~~unsigned~~ int A

+ve

C++/ C#

Java

DIY

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

0 1 1 1 0 1 1 0 1 0 X X X

1234

ans = 0<<1 + 1

= 01<<1 + 1

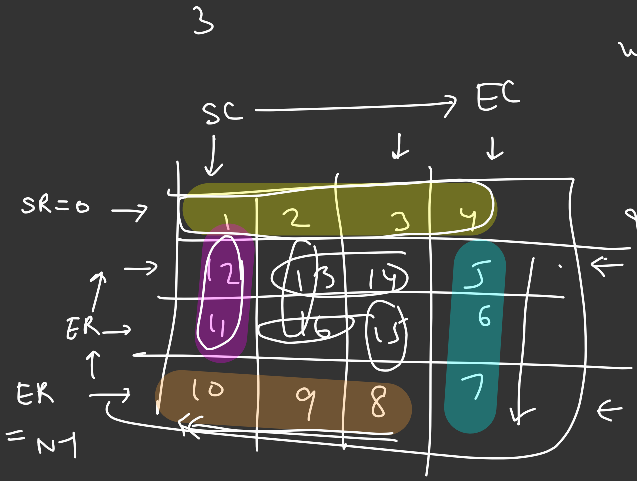
= 010 + 1

= $\boxed{011}$

0110

```
        int
        ↗      long
if( (int)(n&1) == 1){
```

Doubts



3

DIY

```
while( ______ ){
    SR=0    SC → EC          SR++  ⎫
    EC = SR — ER             EC--  ⎬ 4 loop
    ER = EC — SC             ER--  ⎪
    SC = ER — SR             SC++  ⎭
```