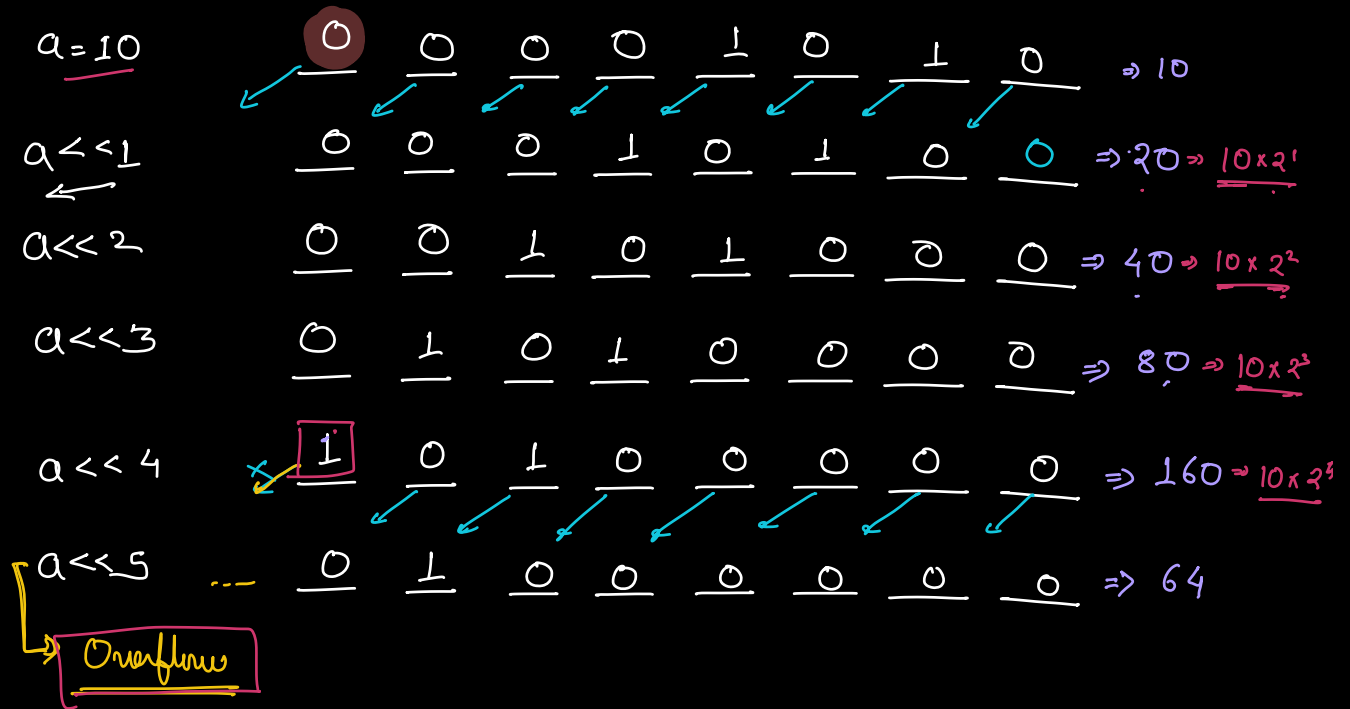


| & ^ ~

left
Shift

right
Shift



$$a \ll n = a \times 2^n$$

(assuming No overflow)

a = 1

$$\begin{aligned}
 a \ll 1 &\Rightarrow 1 \times 2^1 \Rightarrow 2 \\
 a \ll 2 &\Rightarrow 1 \times 2^2 \Rightarrow 4 \\
 a \ll 3 &\Rightarrow 1 \times 2^3 \Rightarrow 8 \\
 a \ll 4 &\Rightarrow 1 \times 2^4 \Rightarrow 16 \\
 &\vdots
 \end{aligned}$$

$$2^n = (1 \ll n)$$

$\Rightarrow a = 20$

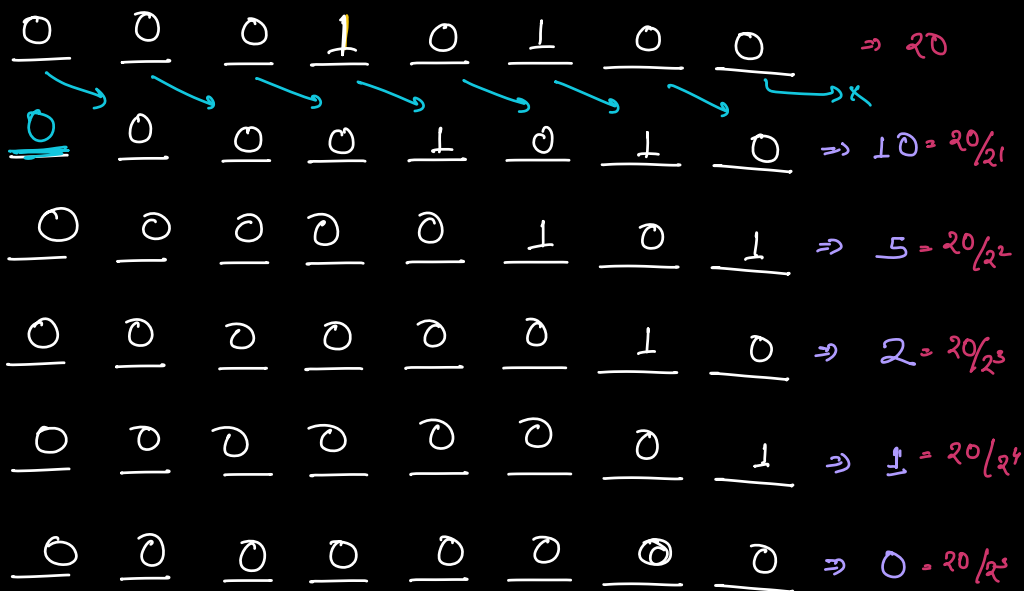
$a \gg 1$

$a \gg 2$

$a \gg 3$

$a \gg 4$

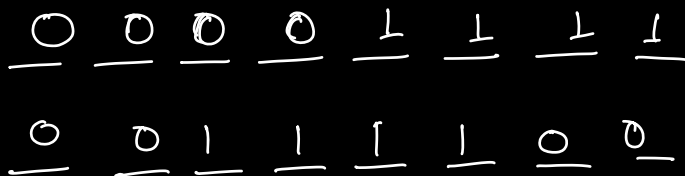
$a \gg 5$



$$a \gg n = a / 2^n$$

Quiz 1

$15 \ll 2$



$$15 \ll 2 = 15 \times 2^2 \Rightarrow 60$$

Quiz 2

$29 \gg 2$

$$= \frac{29}{2^2} \Rightarrow \frac{29}{4} \Rightarrow 7$$

Quiz 3

$$2^n = ?$$

$$a \ll n = a \times 2^n$$

$$a = 1$$

$$1 \ll n = 2^n$$

Quiz 4

$$5^n = ?$$

$$5 * (1 \ll n) \Rightarrow 5 \times 2^n \quad \times$$

$$5 \ll n \Rightarrow 5 \times 2^n \quad \times$$

$$5 \ll (n-1) \Rightarrow 5 \times 2^{(n-1)} \quad \times$$

Q

Given two positive no. N & i .

Check if i th bit in N is set or unset.

1 \rightarrow Set bit

0 \rightarrow Unset bit

$$N = 4 \quad \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{array}$$

$$i = 1$$

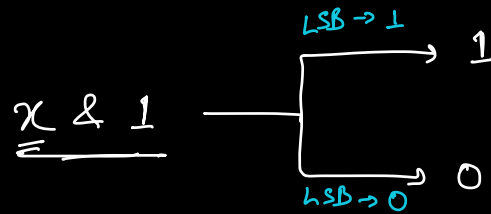
\Rightarrow false

$$N = 10 \quad \begin{array}{ccccccc} & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array}$$

$$i = 2 \longrightarrow \text{false}$$

$$i = 3 \longrightarrow \text{true}$$

Check if the right most bit is set / unset?



$$N : \begin{matrix} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{matrix}$$

$$\begin{aligned} N \gg 1 & \quad 00000101 \\ N \gg 2 & \quad 00000010 \\ N \gg 3 & \quad 00000001 \end{aligned}$$

$$(N \gg i) \& 1$$

 Brings i^{th} bit at right most posn (LSB)

$$\begin{aligned} & \begin{matrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ \Rightarrow & 0 & 0 & 1 & 0 & 0 & 0 \end{matrix} \leftarrow (1 \leq i) \\ & \underline{\hspace{1cm}} \\ & \begin{matrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{matrix} \Rightarrow 8 \end{aligned}$$

$$N \& (1 \ll i) \rightarrow \begin{cases} i \rightarrow 0 & 0 \\ i \rightarrow 1 & 1 \end{cases}$$

```

boolean checkBit (N, i)
// Right shift N by i
if ((N >> i) & 1 == 1)
    return true;
else
    return false;
  
```

```

boolean checkBit (N, i) {
// 1 left shift by i
if ((N & (1 << i)) == 0)
    return false;
else
    return true;
}
  
```

$$TC : O(1) \quad | \quad SC : O(1)$$

Q Given two no. N & i .

Set the i^{th} bit in N .

$\begin{cases} 0 \Rightarrow 1 \\ 1 \Rightarrow 1 \end{cases}$

$N = 4$

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	1	0

$i = 1$

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	1	0

$\Rightarrow 6$

$i = 0$

$\Rightarrow 5$

$i = 2$

$\Rightarrow 4$

int setBit(N, i) {

set $N \mid (1 \ll i);$

}

$0 \mid 1 \rightarrow 1$
 $1 \mid 1 \rightarrow 1$

	7	6	5	4	3	2	1	0
	0	1	1	0	0	1	0	
1	0	0	0	0	1	0	0	
<hr/>								
	0	1	1	0	1	1	0	

$N =$

	7	6	5	4	3	2	1	0
	0	0	0	0	1	0	1	0
	0	0	0	0	0	1	0	0
$i = 2$	<hr/>					1	1	0

$(1 \ll i)$

$(1 \ll i)$

000010000

HW: Clear / unset the i^{th} bit

$0 \rightarrow 0$
 $1 \rightarrow 0$

Amazon Q Toggle the i^{th} bit of N .

$0 \longrightarrow 1$

$1 \longrightarrow 0$

$N = 10$

7 6 5 4 3 2 1 0
0 0 0 0 1 0 1 0

$i = 2$

0 0 0 0 1 1 1 0 $\Rightarrow 14$

$i = 1$

0 0 0 0 1 0 0 0 $\Rightarrow 8$

$0 \wedge 1 = 1$
$1 \wedge 1 = 0$

7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0
0	0	0	0	1	0	0	0
<hr/>							
0	0	0	0	0	0	1	0

$a \wedge 0 = a$

$1 \wedge 0 = 1$
 $0 \wedge 0 = 0$

int toggleBit(N, i) {

 ret ($N \wedge (1 \ll i)$);

}

Tower Research

Q Given a no N . Toggle all the bits starting from the right most set bit.

$N = 20 \Rightarrow 10 \underline{1} 00$
 $\Rightarrow 10011$

$N = 24 \Rightarrow 1 \underline{1} 000$
 $\Rightarrow 10111$

for i starting from 0

if (checkbit(N , i) == false) {
 setBit(N , i);

}

else { toggleBit(N , i);

break;

}

}

return toggleRgBit(N);

return($N-1$);

}

5

3 : 011

$S : 101$

15 : 1111

6 : 110

8 : 1000

20 : 10 100

$$24 \div 11000$$

44 ; 101100

0 : 0000

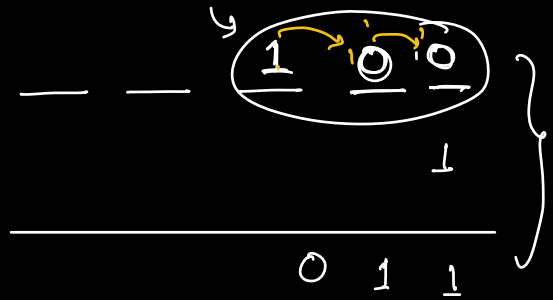
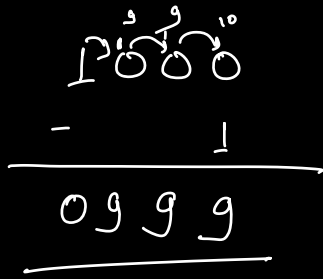
$$(N-1)$$
$$010 \Rightarrow 2$$
$$100 \Rightarrow 4$$
$$1110 \Rightarrow 14$$
$$101 \Rightarrow 5$$
$$0111 \Rightarrow 7$$
$$10011 \Rightarrow 19$$
$$\underline{10111} \Rightarrow 23$$
$$101011 \Rightarrow 43$$
 κ is odd

a b c d 1

a b c d 0 ↓

 x is even

$$\begin{array}{r} 1 \\ \hline 01 \end{array}$$



Google
Amazon

Q Given an int N Count the no. of set bits in N.

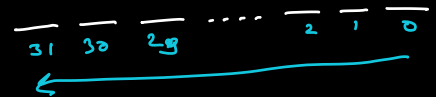
$$N = 10 \Rightarrow 2$$

ten

$$N = 8 \Rightarrow 1$$

$$N = 15 \Rightarrow 4$$

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



$$\underbrace{000\ 000\ 000}_{N \& 1}$$

int countSetBits (N) {

int ans = 0;

while (N > 0) {

if ((N & 1) == 1) {

ans ++;

}

$N = N \gg 1;$ $\Rightarrow N = N/2$

}

return ans;

}

$$N \rightarrow \frac{N}{2} \rightarrow \frac{N}{4} \rightarrow \frac{N}{8} \dots 0$$

$\underbrace{\hspace{10em}}_{\log_2 N}$

2	27	
2	13	1
2	6	1
2	3	0
2	1	1
2	1	1

$\log_2 N$

TC: $O(\log N)$

$$\begin{array}{cccc|c} 1 & 0 & 0 & 0 & \leftarrow \\ 0 & 0 & 0 & & \\ \hline \end{array}$$

N	$(N-1)$	$N \& (N-1)$
3 : 011	010 \Rightarrow 2	010
5 : 101	100 \Rightarrow 4	100
15 : 1111	1110 \Rightarrow 14	1110
6 : 110	101 \Rightarrow 5	100
8 : 1000	0111 \Rightarrow 7	0000
20 : 10100	10011 \Rightarrow 19	10000
24 : 11000	10111 \Rightarrow 23	10000
44 : 101100	101011 \Rightarrow 43	101000

$$1 \& 0 \Rightarrow 0$$

$$0 \& 1 \Rightarrow 0$$

$$a \& (\sim a) = 0$$

$N \& (N-1) \rightarrow$ Unset the right most set bit

1 0 1 1 0 1 0
 $\downarrow N \& (N-1)$
 1 0 1 1 0 0 0
 \downarrow
 1 0 1 0 0 0 0
 \downarrow
 1 0 0 0 0 0 0
 \downarrow
 0 0 0 0 0 0 0

of iteration = No of set bit in the given No.

$$\begin{aligned}
 15 &= 1111 \Rightarrow 4 \\
 16 &= 10000 \Rightarrow 5 \\
 17 &= 10001 \Rightarrow 5 \\
 2^{32} &= 1000\dots00 \Rightarrow 1
 \end{aligned}$$

$$\log_2 15$$

$$\log_2 16 = 4$$

$$\log_2 17$$

$$\log_2 2^{32} \Rightarrow 32$$

```
int countSetBits (N) {
```

```
    int ans = 0;
```

```
    while (N > 0) {
```

```
        N = N & (N-1);
```

```
        ans++;
```

```
    }
```

```
    return ans;
```

```
}
```

TC : $O(\log N)$