Bitwise Operators :-

& | ^ ~ << >>

truth table

a	b	&	|	^ (XOR (exclusive OR))	~ (on a)
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

byte a :- 28 :- 0 0 0 1 1 1 0 0

byte b :- 18 :- 0 0 0 1 0 0 1 0

a & b :- 0 0 0 1 0 0 0 0 ⇒ 16

a | b :- 0 0 0 1 1 1 1 0 ⇒ 30

a ^ b :- 0 0 0 0 1 1 1 0 ⇒ 14

~a :- 1 1 1 0 0 0 1 1 ⇒ -29

↓

1's a

$-2^7 + 2^6 + 2^5 + 2^1 + 2^0$

$= -128 + 64 + 32 + 2 + 1$

$\Rightarrow -128 + 99$

$\Rightarrow -29$

⇒ Odd | Even check

int a = 10 ⇒ ---- 1 0 1 0

int x = 1 ⇒ ---- 0 0 0 1

a & 1 ⇒ 0 0 0 0 ⇒ 0

int a = 11 ⇒ ---- 1 0 1 1

int x = 1 ⇒ ---- 0 0 0 1

a & 1 ⇒ 0 0 0 1 ⇒ 1

any odd ⇒ ---- 1 (odd)

any odd → a & 1

2^x ⇒ even

even → 2^0 → 1

any even ⇒ ---- 0

even

: for any odd no. a, a & 1 == 1.

: for any even no. a, a & 1 == 0.

```
if( a & 1 == 0){
    // a is even
}
```

```
if(a & 1 == 1){
    // a is odd
}
```

: Properties

i) $a \& 0 = 0$.

ii) $a \& a = a$

iii) $a | 0 = a$

iv) $a | a = a$.

v) $a \wedge 0 = a$

vi) $a \wedge a = 0$

vii) $a \& b = b \& a$

viii) $a | b = b | a$

ix) $a \wedge b = b \wedge a$

x) $a \& b \& c$
$= c \& b \& a$
$= b \& c \& a$
$= a \& c \& b$

xi) $a | b | c$
$= b | a | c$
$= c | a | b$
$= c | b | a$

xii) $a \wedge b \wedge c$
$= c \wedge b \wedge a$
$= b \wedge c \wedge a$
$= c \wedge a \wedge b$

Q1. $a \wedge b \wedge b \wedge c \wedge a$

$\Rightarrow a \wedge a \wedge b \wedge b \wedge c$

$\Rightarrow 0 \wedge b \wedge b \wedge c$

$\Rightarrow b \wedge b \wedge c$

$\Rightarrow 0 \wedge c$

$\Rightarrow c.$

Q2. $c \wedge d \wedge a \wedge c \wedge a \wedge g \wedge d \Rightarrow g$

$\Rightarrow c \wedge c \wedge a \wedge a \wedge d \wedge d \wedge g$

$\Rightarrow 0 \wedge a \wedge a \wedge d \wedge d \wedge g$

$\Rightarrow a \wedge a \wedge d \wedge d \wedge g$

$\Rightarrow 0 \wedge d \wedge d \wedge g$

$\Rightarrow d \wedge d \wedge g$

$\Rightarrow g$

Q3. Given N arr elements, every element repeats itself twice except any one unique element, find that unique element.

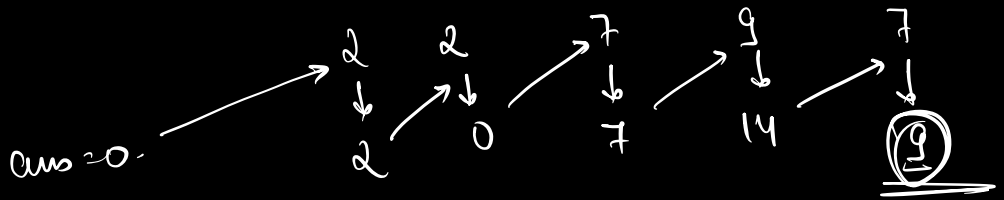
eg $\Rightarrow$ arr[7] $\Rightarrow$ 3 8 4 8 3 7 4

O/P $\Rightarrow$ 7.

ex ⇒ arr[5] ⇒ 2 2 7 9 7

o/p = 9.

⇒ 2 ^ 2 ^ 7 ^ 9 ^ 7 ⇒ 9



9 ⇒ 1 0 0 1

7 ⇒ 0 1 1 1

1 1 1 0

2^3 2^2 2^1 2^0

8 4 2 1

14 ⇒ 1 1 1 0

7 ⇒ 0 1 1 1

9 ⇒ 1 0 0 1

Pseudo

```
uniqueElement(arr[]) {
    ans = 0
    for(i=0; i<N; i++) {
        ans = ans ^ arr[i];
    }
    return ans;
}
```

TC ⇒ O(N)

SC ⇒ O(1)

: repeats thrice :

a ^ b ^ c ^ c ^ b ^ a ^ g ^ a ^ c ^ b.

g ^ a ^ c ^ b

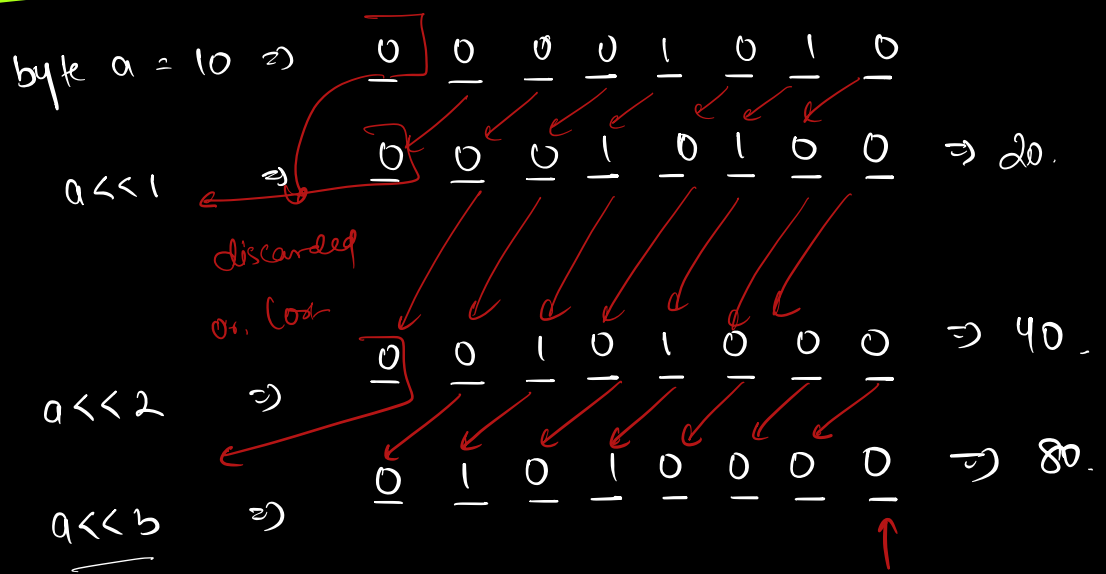
: repeats 4 times :

a ^ b ^ g ^ b ^ a ^ a ^ b ^ b ^ a

g

: SDA only works if the nos. are repeated even no. of times, won't work for odd no. of repetitions.

Leftshift :



$a = 10.$

$a<<1 \Rightarrow 20 \Rightarrow 10 * 2^1$

$a<<2 \Rightarrow 40 \Rightarrow 10 * 2^2$

$a<<3 \Rightarrow 80 \Rightarrow 10 * 2^3.$

⋮

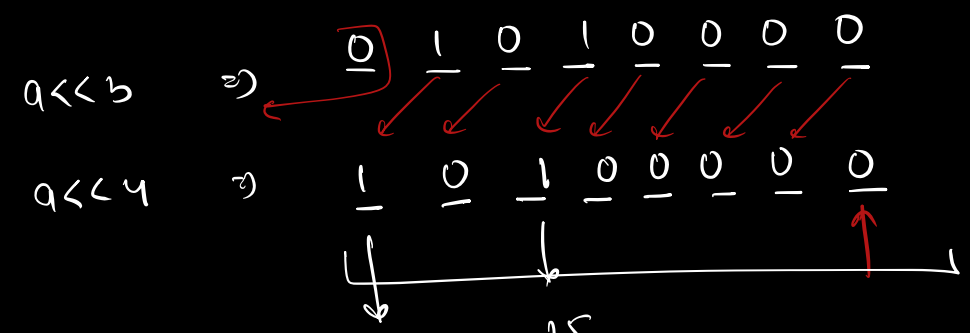
$a<<N \Rightarrow a * 2^N$

$a<<4 \Rightarrow a * 2^4$

$\Rightarrow 10 * 16$

$\Rightarrow 160$

⇒ not possible for 8-bits [overflow]



$-27 + 25$

$= -128 + 32 \Rightarrow -96$

$a << N = a * 2^N$ [by no overflow happens]

Q4. Find 2^N.

2^N

$pow(N) \Rightarrow$ power function

$pow(N) \Rightarrow$ loop.

$1 * 2^N$

$a * 2^N$

1

$a << N \Rightarrow a * 2^N$

$\downarrow$

1

Ans $\Rightarrow 1 << N$.

return $1<<N$ → TC $\Rightarrow O(1)$ SC $\Rightarrow O(1)$

if t is int int $[-2^{31}, 2^{31}-1]$

$N = 29 \rightarrow 2^{29}$ (✓)

$N = 30 \rightarrow 2^{30}$ (✓)

$N = 31 \rightarrow 2^{31}$ (✗✗)

if t is long $[-2^{63}, 2^{63}-1]$

$N = 40 \rightarrow 2^{40}$ (✓)

$N = 60 \rightarrow 2^{60}$ (✓)

$N = 63 \rightarrow 2^{63}$ (✗)

if (N > 62)

Python → no issue → impl. by Strings

Java → Big Integer [Ub Jun → no limit]

Js → range(x).

C → more to C++ (Java.

// calculate 5^N

$5 << N$

↓

$5 << N$ (XX wrong)

for loop, power fun.

$a << N = a * 2^N$

$5 << N \Rightarrow 5 * 2^N$

$a << N$

$= a * 2^N$

$5 << N$

↓

$5 * 2^N = 5^N$

$5 * 2^N$

$5 * 2^4$

$= 5 * 16$

$= 80$

5^N & $N = 4$

5^4

$= 5 * 5 * 5 * 5$

$= 25 * 25$

$= 625$

$a \rightarrow a * 2^N$	$a << N$

5^N $\quad 5 \neq 2^N$

7^N $\quad 7 \neq 2^N$

11^N

13^N

6^N $\quad 6 \neq 2^N$

$5 \neq 2^N \Rightarrow 5 << N.$

$10 \neq 2^N \longrightarrow 10 << N.$

$13 \neq 2^N \Rightarrow 13 << N$

$a << N = a \neq 2^N.$

$\Rightarrow a = 5$

$\Rightarrow 5 << N \Rightarrow 5 \neq 2^N \neq 5^N$

8:55 AM ⇒ 6 mins.

$4^N \Rightarrow (2^N)^2 \Rightarrow (2^2)^N$

$\downarrow$

$(1 \ll 2^N)^2$

$\downarrow$

$(1 << N)^2$.

! Right Shift :-

byte a = 50 ⇒ 0 0 1 1 0 0 1 0

a >> 1 ⇒ 0 0 0 1 1 0 0 1 ⇒ 25 ⇒ 50/2

a >> 2 ⇒ 0 0 0 0 1 1 0 0 ⇒ 12 ⇒ 25/2

a >> 3 ⇒ 0 0 0 0 0 1 1 0 ⇒ 6 ⇒ 12/2

a = 50.

a >> 1 ⇒ 25 ⇒ $\frac{50}{2^1}$

a >> 2 ⇒ 12 ⇒ $\frac{50}{4}$ ⇒ 50/22

$a >> 3 \Rightarrow 6 \Rightarrow \frac{50}{8} \Rightarrow 50/2^3$

$a >> N \Rightarrow \frac{a}{2^N}$ → always

$1 >> N = \frac{1}{2^N}$

$3 >> 2 \Rightarrow \frac{3}{2^2} \Rightarrow 3/4 \Rightarrow 0$

& | ^ ~ << >>

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

$N = 26 \Rightarrow \_\_ \_\_\_\_ 1\ 1\ 0\ 1\ 0$ (bit positions 2 1 0)

$i = 2$

o/p ⇒ false

N = 35 ⇒ ——— 1 0 0 0 1 1
(5 4 3 2 1 0)

i = 1

Q) i = true.

N = 1 0 0 1 0 1 1 0
(7 6 5 4 3 2 1 0)

i = 0 → N & 1 ⇒ 0

(N & 1 == 1)
return true
else return false

i = 1 ⇒ if((N >> 1) & 1 == 1)
return true
else return false.

TC ⇒ O(1)
SC ⇒ O(1)

Ans

```
boolean checkBit(N, i){
    if((N >> i) & 1 == 1)
        return true
    else return false.
```

Doubts

$4^N \Rightarrow (1<<N)^2$

$\downarrow$

$(2^2)^N$

$\Rightarrow (2^N)^2$

$\Rightarrow (1<<N)^2$

```
x = 1<<N
return x * x
```

2^{2N}

$\downarrow$

$1 << (2N)$

$1 * (2)^{2N}$

---

i :=

$n \& 1 == 0$ | even → last 0.

$n \& 1 == 1$ | odd → last 1

① 1101 & 0001 ← = 0001 = 1
