Today's agenda

- Binary number system
- operators
- Problems
- Constraints

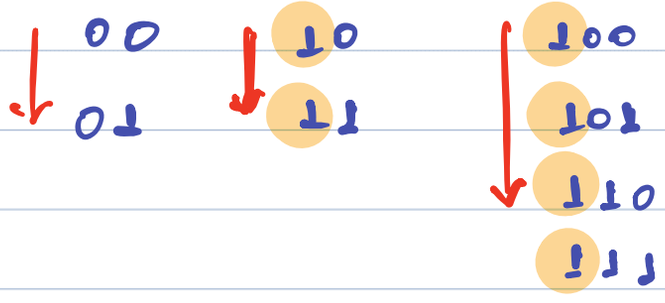
↳ decimal no. system

↳ {0-9} system

0	10	20	...	90	-
1	11	21		91	
2	.	.		92	
3				.	
4					
.					
.					
.					
9	19	29		99	

↳ binary no. system

↳ {0 1}



Conversion

Convert decimal to binary:

ex: 30

2	30	— 0
2	15	— 1
2	7	— 1
2	3	— 1
2	1	— 1
	0	

→ 11110

Quiz:

2	45	1
2	22	0
2	11	1
2	5	1
2	2	0
2	1	1
	0	

101101

Ub binary no. to decimal no. → bit index

ex: $(1\ 0\ 1\ 0\ 1)_2$ (bit indices 4 3 2 1 0; 2^3*0, 2^1*0)

1^4*1, 1^2*1, 2^0*1

0: unset bit / off bit

1: set bit / on bit

→ $2^4*1 + 2^3*0 + 2^2*1 + 2^1*0 + 2^0*1$

$16 + 0 + 4 + 0 + 1 = 21$

Quiz:

6 5 4 3 2 1 0

1 0 1 1 0 1 0

Ub $2^1 + 2^3 + 2^4 + 2^6 = 90$

$2 + 8 + 16 + 64$

→ $(1\ 0\ 1\ 0\ 2\ 0)_2$: invalid input

4. Add binary numbers

```
  1 1
  3 6 8
  4 5 4
-------
  8 2 2
```

0 + 0 → 0

0 + 1 → 1

1 + 0 → 1

1 + 1 → 10

1 + 1 + 1 → 11

①

```
  +1 +1 +1 +1
  0  1  0  1  1  1
  0  1  1  1  1  0
-----------------
  1  1  0  1  0  1
```


Quiz

```
  +1    +1 +1 +1 +1
  0  1  0  1  0  1  1
  0  1  0  0  1  1  1
-------------------
  1  0  1  0  0  1  0
```

* int → 32 bits

  long → 64 bits

Bitwise operators: { and, or, xor, leftShift, rightShift }

A	B	A & B	A | B	A ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

0 dominant

1 dominant

Same same
Puppy shame

Symbol → &&

relational op.

```
if (n%2 == 0 && n%3 == 0){

}
```

false

|| → relational op.

true dominant

& → bitwise operator

0 dominant

| → bitwise operator

1 dominant

(i) 23 & 10 = 2

```
      4 3 2 1 0
23:   1 0 1 1 1
10:   0 1 0 1 0
      ---------
      0 0 0 1 0   → 2^1 = 2
```

Quiz:

```
20:   1 0 1 0 0
10:   0 1 0 1 0
      ---------
      0 0 0 0 0   → 0
```

(ii) 23 | 10

```
      4 3 2 1 0
23:   1 0 1 1 1
10:   0 1 0 1 0
      ---------
      1 1 1 1 1   → 2^0 + 2^1 + 2^2 + 2^3 + 2^4 = 2^5 - 1 = 31
```

Quiz:

```
      4 3 2 1 0
20:   1 0 1 0 0
10:   0 1 0 1 0
      ---------
      1 1 1 1 0   → 2^1 + 2^2 + 2^3 + 2^4 = 30
```

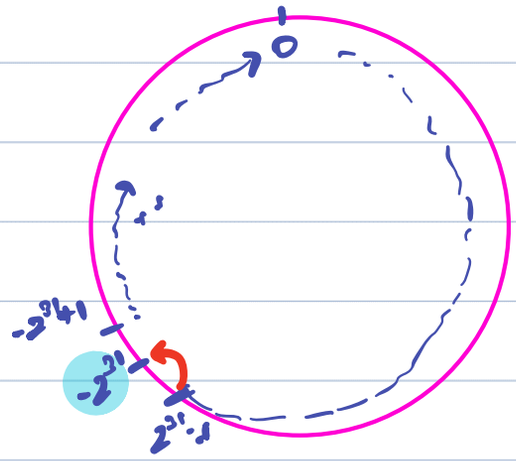
(iii) 23 ^ 10

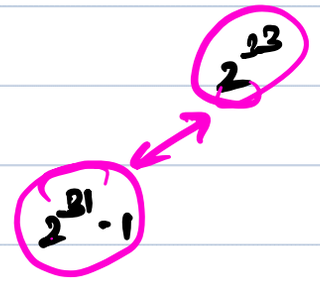
```
      4 3 2 1 0
23:   1 0 1 1 1
10:   0 1 0 1 0
      ---------
      1 1 1 0 1   → 29
```

Quiz:

```
      4 3 2 1 0
20:   1 0 1 0 0
15:   0 1 1 1 1
      ---------
      1 1 0 1 1   → 2^0 + 2^1 + 2^3 + 2^4 = 27
```

int → 32 bits → -2^{31} to $2^{31}-1$





$2^{32}-2^{31}$

$2^{31}(2^1-1) = 3 * 2^{31}$

$2^{31}-1+1 = -2^{31}$

$\frac{1}{2^{31}}$

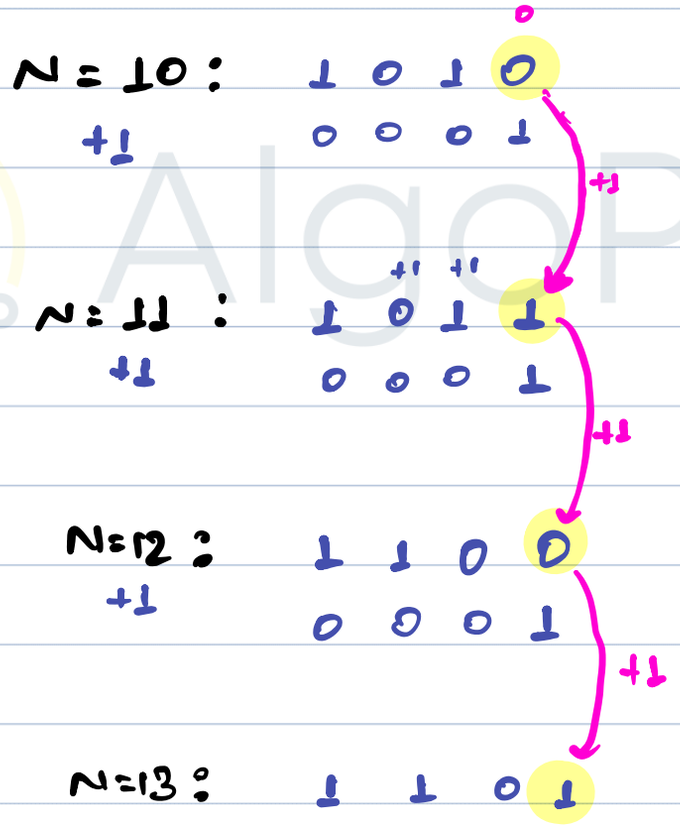
Break till 10:40 pm

Q) you have been given a Positive no., identify whether the number is even or odd.

Ex: N = 8 → even

N = 7 → odd

Note: use of +, −, *, / or % is not allowed.



by 0th bit index for even no. will be 0.

by 0th bit index for odd no. will be 1.

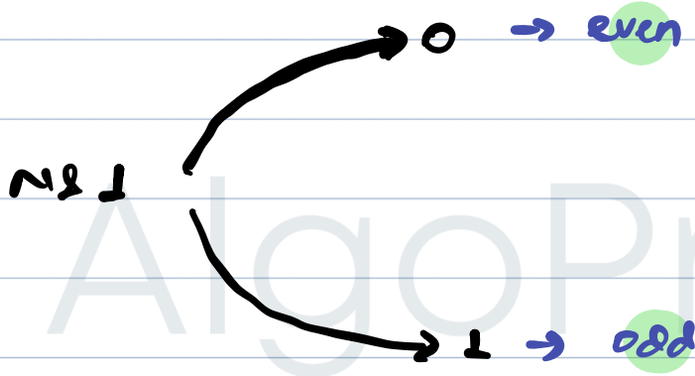
How to check last bit?

int N = 10 :

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

N = 11 :

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




```
void checkeven (int n){
    if (n & 1 == 0){
        s.o.p("Even");
    }
    else{
        s.o.p("odd");
    }
}
```

Properties

(i) Commutative Property

$a \& b = b \& a$

$a | b = b | a$

$a \hat{} b = b \hat{} a$

(ii) Associative Property

$a \& b \& c \rightarrow (a \& b) \& c = a \& (b \& c)$

$a | b | c \rightarrow (a | b) | c = a | (b | c)$

(iii)

(a) $N \& N = N$

$N \rightarrow 101$
$\&$
$N \rightarrow 101$
101

$N \rightarrow 101$
$\&$
$0 \rightarrow 000$
000

(b) $N \& 0 = 0$

(c) $N | 0 = N$

$N \rightarrow 101$
$|$
$0 \rightarrow 000$
101

$N \rightarrow 101$
$|$
$N \rightarrow 101$
101

(d) $N | N = N$

(e) $N \hat{} 0 = N$

$N \rightarrow 101$
$\hat{}$
$0 \rightarrow 000$
101

$N \rightarrow 101$
$\hat{}$
$N \rightarrow 101$
000

(f) $N \hat{} N = 0$

Q) Given arr[N], every element appears twice except for one element which appears once, find that unique element.

Eg: arr[7]: { 6 8 8 7 7 10 6 } → 10

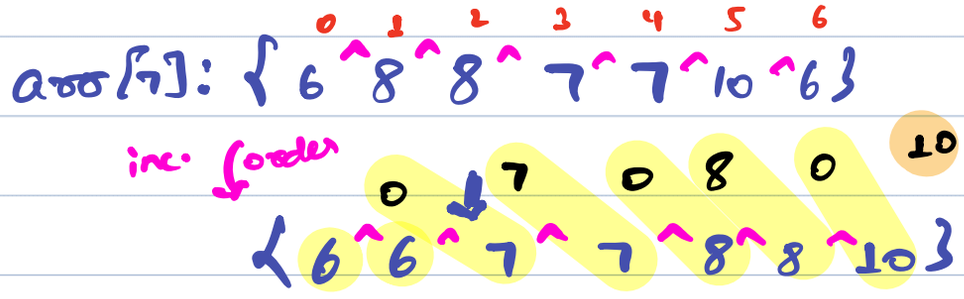
arr[5]: { 2 1 9 2 9 } → 1

Idea1

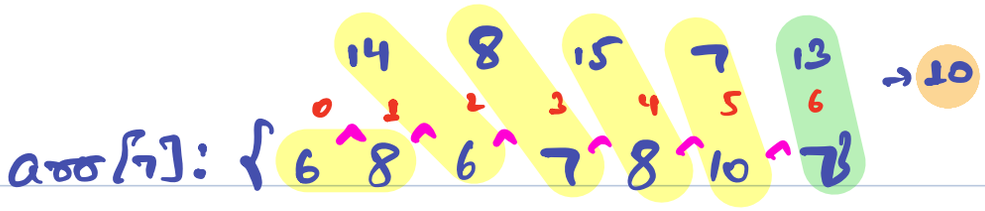
↳ nested loop.

T.C: $O(N^2)$

Idea2



↳ you don't actually need to rearrange.



6: 0 1 1 0
8: 1 0 0 0
1 1 1 0 : 14

14: 1 1 1 0
6: 0 1 1 0
1 0 0 0 : 8

8: 1 0 0 0
7: 0 1 1 1
1 1 1 1 → 15

15: 1 1 1 1
8: 1 0 0 0
0 1 1 1 → 7

7: 0 1 1 1
10: 1 0 1 0
1 1 0 1 → 13

13: 1 1 0 1
7: 0 1 1 1
1 0 1 0 → 10

//Pseudo code

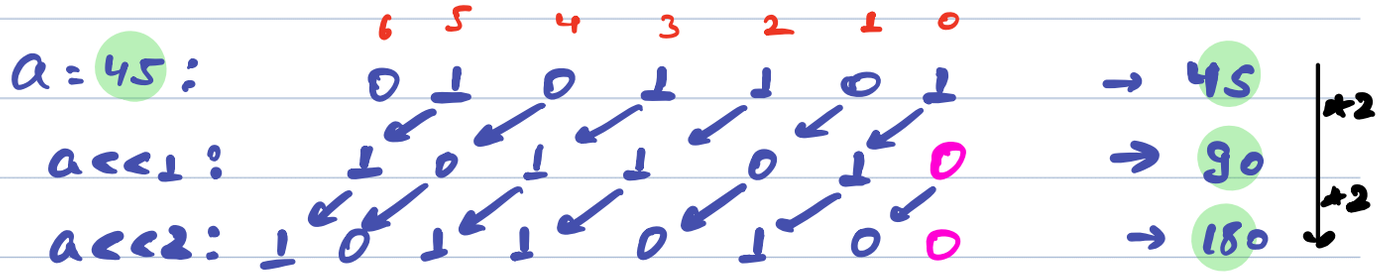
```
int uniqueelement (int arr[n]){

    int ans = 0;
    for (int i=0; i<n; i++){
        ans = ans ^ arr[i];
    }

    return ans;
}
```

T.C: O(n)
S.C: O(1)

Left Shift (<<) → O(1)



$a << 1 \rightarrow a * 2$

$a << 2 \rightarrow a * 2 * 2$

$a << 3 \rightarrow a * 2 * 2 * 2$

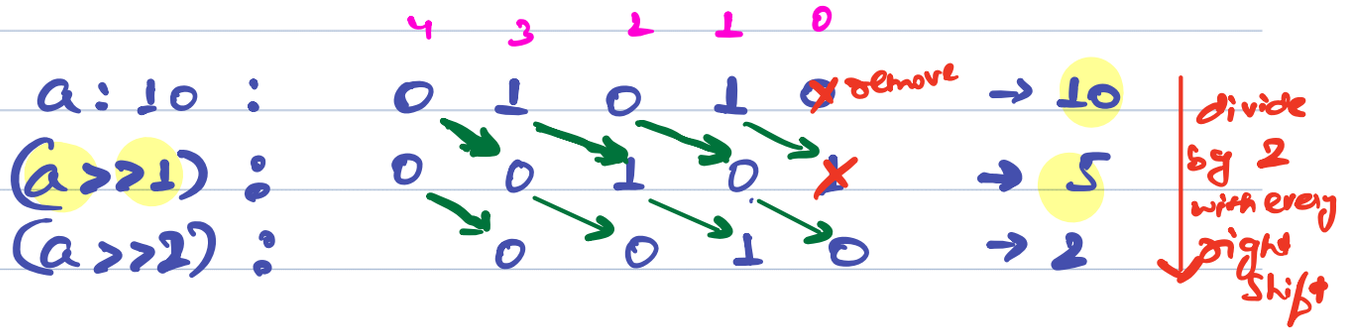
$a << N \rightarrow a * 2^N$

Qui2:

$a = 1$

$1 << N \rightarrow 1 * 2^N = 2^N$

Ls right shift (>>) → O(1)



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

Constraints

1 sec = 10^8 iterations

↳ array length : 10^5

↳ $O(N^2)$ → $(10^5)^2 = 10^{10}$ iterations

↳ $O(N\sqrt{N})$ → $10^5 * \sqrt{10^5} \Rightarrow 10^5 * 10^{2.5} = 10^{7.5}$