19/5/2023

Revision class - 2

Today's quote

The only place where success comes before work is in the dictionary.

Binary no $\{0, 1\}$

OR AND XOR Left shift Right shift.

A	B	OR	AND	XOR → Sum without carry
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

$01 + 01 = 10$ (carry)

$0 + 0 = 0$

$1 + 0 = 1$

$0 + 1 = 1$

Same same puffy shame.

20 & 13 →

2^4	2^3	2^2	2^1	2^0
1	0	1	0	0
& 0	1	1	0	1
0	0	1	0	0 ⇒ 4

20 & 13 → 4

Left shift

int → 4 bytes (32 bits), assume → 8 bits.

7(2^7) 6(2^6) 5(2^5) 4(2^4) 3(2^3) 2(2^2) 1 0

20 = 0 0 0 1 0 1 0 0

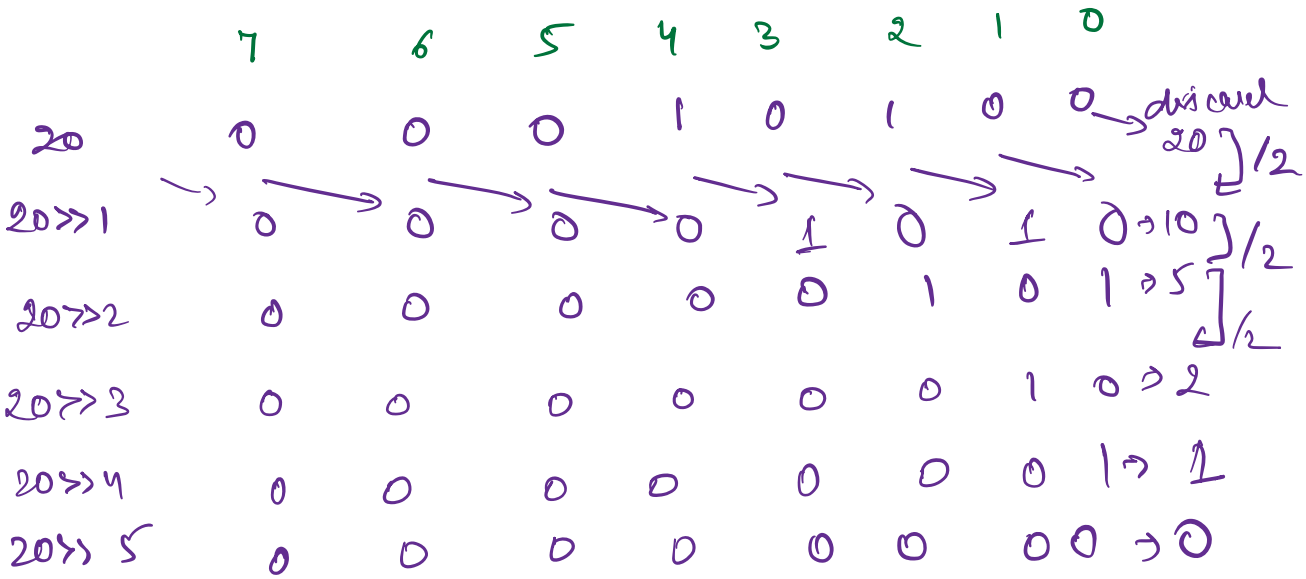
20<<1 = 0 0 0 1 0 1 0 0 = 40 ×2

20<<2 = 0 0 1 0 1 0 0 0 = 80 ×2

20<<3 = 0 1 0 1 0 0 0 0 = 160 ×2

20<<4 = 1 0 1 0 0 0 0 0 ⇒ 64

Right shift


7 6 5 4 3 2 1 0
20 0 0 0 1 0 1 0 0 → discard
20] /2
20>>1 0 0 0 0 1 0 1 0 → 10] /2
20>>2 0 0 0 0 0 1 0 1 → 5] /2
20>>3 0 0 0 0 0 0 1 0 → 2
20>>4 0 0 0 0 0 0 0 1 → 1
20>>5 0 0 0 0 0 0 0 0 → 0

Q, set ith bit of a no.

```
  4 3 2 1 0
  1 0 1 1 0      ,   i = 3
OR 0 1 0 0 0
-------------
  1 1 1 1 0
```

Q Toggle ith bit

```
    4 3 2 1 0
    1 0 0 1 0      i = 2nd
^   0 0 1 0 0
-------------
    1 0 1 1 0
```

Q Generate a number [x is followed by y 0s].

x = 3, y = 4 → 1110000

x = 2, y = 1 → 110

① Run a loop

111 000

x , x << 1 + 1

x = 3, 111 → 7 ($2^3 - 1$)

x = 2, 11 → 3 ($2^2 - 1$)

x = 4, 1111 → 15 ($2^4 - 1$)

$(2^x - 1)$

1 << x

2^x

1 << 1

2

1

10

((1<<x) - 1) << y → y zeros

111000

x ones

x = 3, y = 4 → 1110000

1000

$(2^3 - 1)$ → 1<<3 → 1000 - 0001 → 111

((1<<3) - 1) → 111 → A<<y

A

111 << 1 → 1110

111 << 2 → 11100

111 << 3 → 111000

$x = 5, y = 8$

11111,00000000

$(2^5 - 1) << 8$

$(1 << (x+y) - 1) \wedge ((1 << y) - 1)$

$x = 3, y = 4$

$1 << 4$

$10000 - 1 \Rightarrow 01111$

10000000

01111111

00011111

11110000

Q. count set bits.

$11001 \Rightarrow 3$, $100 \Rightarrow 1$, $0 \Rightarrow 0$

$11001 \;\&\; 00001 = 1$

$1100 \;\&\; 0001 = 0$

$110 \;\&\; 001 = 0$

$11 \;\&\; 01 = 1$

$1 \;\&\; 1 = 1$

$0 \;\&\; 1 = 0$

```
while (x > 0) {
    ans += (x & 1);   →  if (x & 1 == 1) ans++
    x >>= 1;
}
```

Help from Sam

Target A.

Alex start with 0. He can double his score by a doing a question. / can take help from friend & get a score of 1 added.

A = 5 → 0 → 1 → 10 → 100 → 101 → 2

↓

101 →

A = 3 → 0 → 1 → 10 → 11 → 2

11

A = 7 → 0 → 1 → 10 → 11 → 110 → 111 → 3

111

A = 2 → 0 → 1 → 10 → 1

10

15 → 0 → 1 → 10 → 11 → 110 → 111 → 1110 → 1111

1111 → 4

Sorting

→ arrangement of data in a particular order on the basis of some parameter.

Comparator

9	3	10	6	4	→	3	4	9	6	10
↓	↓	↓	↓	↓		↓	↓	↓	↓	↓
3	2	4	4	3		2	3	3	4	4

Sort() → asc order of magnitude.

Sort (start, end, comp)

→ rules of sorting.

count(x) < count(y)
x should come first

count(x) > count(y)
y should come first

count(x) == count(y)
x <= y
x should come first
x > y
y should come first

if first arg should come first return true.

else return false.

```
bool comp (int x, int y) {
    int cntx = count_factor(x);
    int cnty = count_factor(y)
    if (cntx < cnty)
        return true;
    else if (cntx > cnty)
        return false.
    else {
        if (x <= y)
            return true;
        else
            return false;
    }
}
```

Q. Ten digit sorting.

Given array A of N integers. Sort the array in increasing order of the value at the tens place of every no.

- If a no has no tens digit, assume it to be 0
- If 2 nos have same tens digit, sort descending.

A: [15, 11, 7, 19] → [7, 19, 15, 11]

A: [2, 24, 22, 19] → [2, 19, 24, 22]

```
bool comp(int x, int y){
    int d1 = (x/10)%10;
    int d2 = (y/10)%10;
    if (d1==d2){
        if (x>y){
            return true;
        }
        return false;
    }
```

```
if (d1<d2){
    return true;
}
else{
    return false;
}
}
```

Modular Arithmetic

$$(a+b)\%p = (a\%p + b\%p)\%p$$

$\{0,p-1\}$ $\{0,p-1\} \rightarrow \{2p\} \rightarrow \{0,p-1\}$

$\{0-p-1\}$

a b p $\quad (8+6)\%10 = 8\%10 + 6\%10$

8 6 10 $\Rightarrow$ 4 $\quad = 14\%10 = 4$

$$(a*b)\%p = \left(\left(a\%p\right) \times \left(b\%p\right)\right)\%p$$

$\{0,p-1\} * \{0,p-1\} \Rightarrow p \rightarrow \{0,p-1\}$

Q Given a, n, p, calculate $a^n \% p$, without inbuilt functions.

Constraints $1 \le a \le 10^9$, $2 \le p \le 10^9$ $1 \le n \le 10^5$

Ex1 $a=3, n=4, p \rightarrow 7$, $(3^4)\%7 \Rightarrow (81\%7) = 4$

```
— powmod(int a, int n, int p) {
    long ans = 1;
    for (i = 1; i <= n; i++) {
        ans = (ans * a) % p → [10^9]
    }        [p-1]  [a] ≈ a*p
    return ans % p   ≈ 10^9 * 10^9
}                    ≈ 10^18
```

Issues in code

a	n	p	$(a^n)\%p$
2	30	45	$(2^{30})\%45$
2	60	48	$(2^{60})\%45$
2	100	45	$(2^{100})\%45$

Key sum // a, p, n = 3

ans	i	i<=3	ans = (ans * a) % p
1	1	✓	ans = a % p -> no overflow
a % p	2	✓	ans = (a % p * a) % p

= ((a % p) % p * (a % p)) % p

= ((a % p * (a % p)) % p

ans = (a * a) % p = a^2 % p -> no overflow

0, p-1

a^2 % p	3	✓	ans = (a^2 % p * a) % p

= (a^2 % p % p * a % p) % p

= (a^2 % p * a % p) % p

=> (a^3 % p) => (0, p-1)

Q Given a number in arr[] format, calculate arr[] % p

Constraints

$1 \le N <= 10^5$

$0 \le arr[i] \le 9$

$2 \le p \le 10^9$

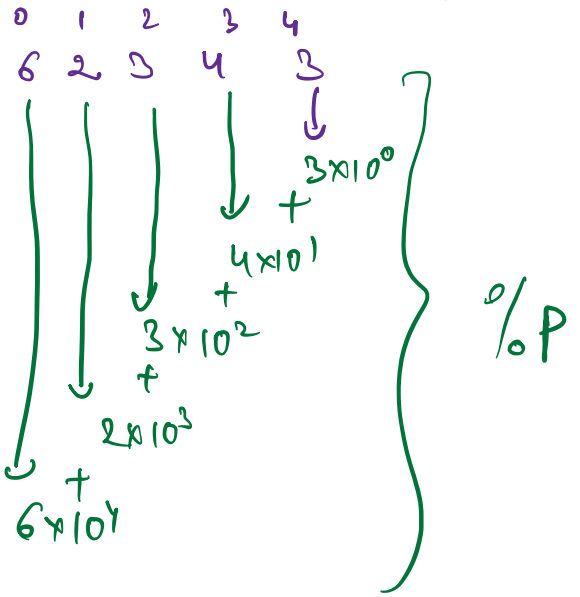
Ex N=5 arr[5] = | 6 | 2 | 3 | 4 | 3 |, p = 49

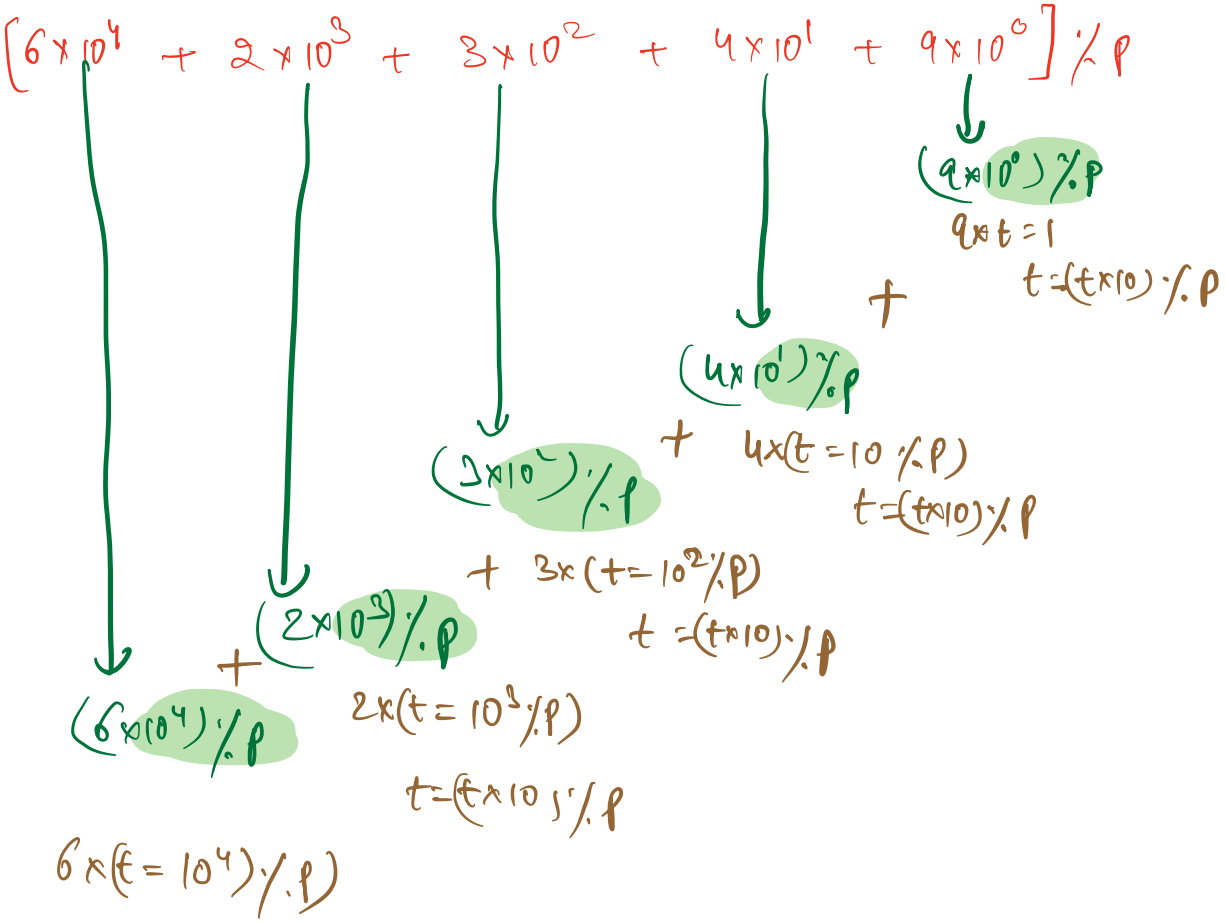
(62343) % 49 = 15

N=4 arr[4] = | 2 | 4 | 3 | 7 |, p = 16

2437 % 16 = 5

N = 5, arr[5] =




[6x10^4 + 2x10^3 + 3x10^2 + 4x10^1 + 9x10^0] %p
(9x10^0) %p
9x t = 1
t = (t x 10) %p
+
(4x10^1) %p
+ 4x(t = 10 %p)
t = (t x 10) %p
(3x10^2) %p
+ 3x(t = 10^2 %p)
t = (t x 10) %p
(2x10^3) %p
+
2x(t = 10^3 %p)
t = (t x 10) %p
(6x10^4) %p
6x(t = 10^4) %p)

Range of integers

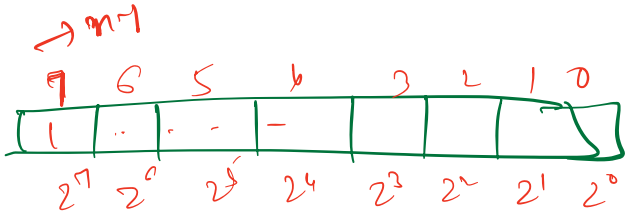
Min no. → -2^{31} → 1 0 0 0 0 0 0 0 . . = -2^{31}

$= -2147483648 \approx -2 \times 10^9$

Max no → 0 1 1 1 1 . . . → $2^{31}-1$

$\Rightarrow 2147483647$

$\simeq 2 \times 10^9$



$-2^7 + 0 + 0 \ldots$

Range of Long

Min no → 1 0 . . . . → $= -2^{63} \approx -9 \times 10^{18}$

-2^{63}

Max no → 0 1 1 . . . $2^{63}-1 \approx 9 \times 10^{18}$

Q Calculate sum of all array elements in the given array.

Constraints $1 \le N \le 10^5$

$1 \le A[i] \le 10^6$

$[10^6 \ldots\ldots] \times 10^5$

10^{11}

```
int/long sum = 0;
for i -> 0 to N-1
    sum += A[i]
return sum;
```

Constraints → TLE

↳ overflow

Q given 2 integer a & b, return a*b;

```
int ans = a*b   X
return ans;
```

```
long ans = (a*b);   X   → overflow at multiplication step.
return ans;
```

```
long ans = long(a*b)   X
```

```
long ans = (long(a))*b
```

b long, b int

long ans = a; ✓
return ans*b; ✓

long ans = 12 * a * b
b
long ✓

doubt

$a^{12} \Rightarrow a^6 * a^6$

$a^{13} \Rightarrow a^6 * a^6 * a$

```
int pow (a, n, p) {
  if (n == 0) { return 1; }
  long X = pow (a, n/2, p) → (a^(n/2)) % p
  if (n % 2 == 0) { return (X*X) % p }
  else { return [(X*X) % p * a) % p],
}
```