Given N, & an array contains all Elements from 1 - N

Except 1 Element find missing elements. ar[N-1]

N = 5 : ar[4] : { 5, 3, 1, 4 } : Ans = 2

1 - 5

N = 6 : ar[5] : { 3, 2, 4, 6, 1 } : Ans = 5

[1 _ 6]

N = 8 : ar[7] : { 2, 3, 8, 4, 7, 6, 1 } : Ans = 5

[1 - 8]

obs:

1) Min Ans : 1   Max Ans : N

Check from 1 - N, The element which is not present

```
i = 1; i <= N; i++) {        → O(N)
    // check i is present in ar[]
}                              ↳ O(N)
```

TC : $O(N^2)$

SC : $O(1)$

→ Iterate & check right?

2) obs: N = 9 [1 - 9] :

↳ ar[8] →

{ 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 } ⇒ $\frac{(9)(10)}{2}$

− { 4, 6, 3, 5, 8, 7, 9, 1, } ⇒ { 2 }

↳ sum of array elements

Generalize

Int range → $[-10^9, 10^9]$

long range → $[-10^{18}, 10^{18}]$

Given N :

Missing Element = Sum of all Elements 1 – N − Sum of all array Elements

$\frac{(N)(N+1)}{2}$

→ TC: O(N) SC: O(1)

Constraints :

$1 \le N \le 10^6$ → $\frac{(N)(N+1)}{2} \approx \frac{(10^6)(10^6+1)}{2} \approx 10^{12}$ x {overflow}

we cannot store in Integer

obs2: $a \wedge a = 0$

$\cancel{a} \wedge b \wedge \cancel{c} \wedge \cancel{a} \wedge \cancel{d} \wedge \cancel{d} \wedge \cancel{d} = b$

Given N = 9 ,

[1 – 9] = {1, 2, 3, 4, 5, 6, 7, 8, 9}

ar[8] = {6, 4, 1, 3, 2, 9, 7, 5}

a = {xor of all Elements 1 – N}

b = {xor of all array Elements}

Missing element = $(a \wedge b)$

b

a = 0

i=1; i<=N; i++

{

a = a ^ i

}

→ $a \wedge b$ TC: O(N) SC: O(1)

b = 0

xor of all Elements

2023

% → Cryptography
→ Hashing

% → Remainder

a % b ⇒ Remainder when a is divided by b

10 % 4 → 2   10 % 6 → 4

Divident = Divisor × Quotient + Remainder

$$\boxed{\text{Remainder} = \text{Divident} - \underbrace{\text{Divisor} \times \text{Quotient}}_{\text{Greatest multiple of Divisor} \le \text{Divident}}}$$

150 % 11 = 150 − (11×13) {143} → 7

100 % 7 = 100 − (98) = 2
7×14

−40 % 7 = −40 − (greatest Multiple of 7 ≤ = −40)
−42

= −40 − (−42) = 2

−60 % 9 = −60 − (greatest Multiple of 9 ≤ = −60)
−63

= −60 − (−63) = 3

Twist

Doubts Session

In C/C++/Java/ vs Python ?

$-40 \% 7 = -5 + 7$

$-60 \% 9 = -6 + 9$

$-30 \% 4 = -2 + 4$

$-40 \% 7 = 2$

$-60 \% 9 = 3$

$-30 \% 4 = 2$

C/C++/Java/ JS/ . — .

$$a \% m = \begin{cases} \text{if } a < 0 \\ \quad a \% m + m \end{cases}$$

Why % Mod ? → { % Mod can limit your data to a given range }

$35 \% 10 \rightarrow 5$

$x \% 10 \rightarrow [0 \_ 9]$

→ Hashing

→ Consistent Hashing

→ HLD

$$\left.\begin{matrix} -\infty \\ \\ +\infty \end{matrix}\right\} \rightarrow \% m \rightarrow \begin{cases} 0 \\ m-1 \end{cases}$$

→ Hashmap

$\% M \rightarrow [0, M-1]$

Modular Arithmetic $\{+, -, /, *\}$

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

$a = 6, b = 8, M = 10$ $6 \% 10$ $8 \% 10$

$6 + 8 = 14$

$(6+8) \% 10 \Rightarrow 4$

1) $(a+b) \% M = (a \% M + b \% M) \% M$

2) $(a*b) \% M = (\underbrace{\overbrace{a \% M * b \% M}}_{\text{Exceed } M}) \% M$

3) $(a-b) \% M = (a/b) \% M$ } Advanced Session TODO

1Q) $\text{power}(a, n, p) \Rightarrow \underbrace{a^n \% p}$

Ex: $a = 2, n = 5, p = 7 \Rightarrow 2^5 \% 7 \Rightarrow 32 \% 7 = 4$

Ex2: $a = 3, n = 4, p = 6 \Rightarrow 3^4 \% 6 \Rightarrow 81 \% 6 \Rightarrow 3$

```
int power(a, n, p) {
    int ans = 1
    i = 1; i <= N; i++) {
        a = a * a
    }
    return a % p
}
```

// Assume no overflow

Constraint : $P \le 10^7$

i	a value Before	a value After
1	a	$a^2 = a^{2^1}$
2	a^2	$a^4 = a^{2^2}$
3	a^4	$a^8 = a^{2^3}$
4	a^8	$a^{16} = a^{2^4}$
N		$a^{2^N} \times a^N$

```
int power(a, n, p) {
    long ans = 1
    i = 1; i <= N; i++) {
        ans = (ans * a) % p
        ans = (ans % p * a % p) % p
    }
    return ans % p }
}
```

$(P-1 \times P-1) \approx (P-1)^2 \approx 10^{18}$

Can we store in int ?

$\{0, P-1\}$

$a = 10, \; N = 40 \quad p = 100$

$\Rightarrow 10^{40} \% 100$

Can we store 10^{40} ✗

TC: O(N) SC: O(1)

10:40pm break

$\Rightarrow (a \% M) \% M = a \% M$

// Eg:

a, N = 5, p → $(a^5) \% p$ →

$(a^4 * a) \% p$. $(a^4 \% p * a \% p) \% p$

$(a^4) \% p$ → $(a^3 * a) \% p$ = $\underbrace{(a^3 \% p * a \% p) \% p}$

//

389) $(3 8 9) \% 3$ → $(3+8+9) \% 3 \neq 0$

$(3 3 6 2 1) \% 3 = (3+3+6+2+1) \% 3 = 0$

Divisibility rule for 3:

→ Sum of digits should be divisible by 3

$(35632) \% 3 =$

$10^4 \; 10^3 \; 10^2 \; 10^1 \; 10^0$

$\{3 * 10^4 + 5 * 10^3 + 6 * 10^2 + 3 * 10^1 + 2 * 10^0\} \% 3$

→

$\{\underbrace{(3 * 10^4) \% 3}_{\downarrow} + (5 * 10^3) \% 3 + (6 * 10^2) \% 3 + (3 * 10^1) \% 3 + (2) \% 3\} \% 3$

$(3 \% 3 * \underbrace{10^4 \% 3}_{1}) \% 3$

$(5 \% 3 * \underbrace{10^3 \% 3}_{1}) \% 3$

$(6 \% 3 * \underbrace{10^2 \% 3}_{1}) \% 3$

$1 \% 3 = 1$

$10 \% 3 = 1$

$10^2 \% 3 = 1$

$10^N \% 3 = 1$

$1 \% 9 = 1$

$10 \% 9 = 1$

$10^2 \% 9 = 1$

$10^N \% 9 = 1$

$(3 \% 3 * 10 \% 3) \% 3$

$(2) \% 3$

$((3 \% 3) + (5 \% 3) + (6 \% 3) + (3 \% 3) + (2 \% 3)) \% 3$

$\Rightarrow (a \% m + b \% m + c \% m + d \% m + e \% m) \% m$

$= (a + b + c + d + e) \% m$

$(3 + 5 + 6 + 3 + 2) \% 3$

Divisible by 9 = (sum of digits divisible by 9)

Divisible by 4 = (last 2 digits should be divisible by 4)

$\Rightarrow (478) \% 4 \Rightarrow (4 * 10^2 + 78) \% 4 \Rightarrow 78 \% 4$

$100 \% 4 = 0$

$10^3 \% 4 = 0$

$10^4 \% 4 = 0$

$\Rightarrow (\underbrace{4 * 10^2 \% 4}_{= 0} + 78) \% 4$

$10^3 \% 8 = 0$

$10^4 \% 8 = 0$

$10^5 \% 8 = 0$

last 3 digits $\% 8 == 0$

Google

Given N & P → int → Calculate N % P

→ Return N % P

→ int ≤ 10^9 → 10 digits → 1 000000 0 000

→ long ≤ 10^{18} } 19 digits

→ N can have 10^5 digits

→ Given string S, P → int → Calculate S % P

1 ≤ S.size() <= 10^5 digits

// Given S_N : S_0 S_1 S_2 S_3 S_4 . S_5 . . S_{N-4} S_{N-3} S_{N-2} S_{N-1}

→ digit

$S_0 * 10^{N-1}$ (N-0-1)

+ $S_1 * 10^{N-2}$ (N-1-1)

+ $S_2 * 10^{N-3}$

+ $S_3 * 10^{N-4}$

+ $S_4 * 10^{N-5}$

+ $S_5 * 10^{N-6}$

+ $S_{N-4} * 10^3$

+ $S_{N-3} * 10^2$

+ $S_{N-2} * 10^1$

+ $S_{N-1} * 10^0$

N-i-1

i

$S_N \% p = (S_0 * 10^{N-1} + S_1 * 10^{N-2} + S_2 * 10^{N-3} + \cdots S_{N-1} * 10^{0}) \% p$

$S \% p = \Big[(S_0 * 10^{N-1}) \% p + (S_1 * 10^{N-2}) \% p + (S_2 * 10^{N-3}) \% p + \cdots (S_{N-1}) \% p \Big] \% p$

$(a * b) \% p$

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

long ans = 0, Given s

```
n = s.size();
i = 0; i < s.size(); i++) {
    ans = (ans + s[i] * power(10, N-i-1, p)) % p
}
return ans;
```

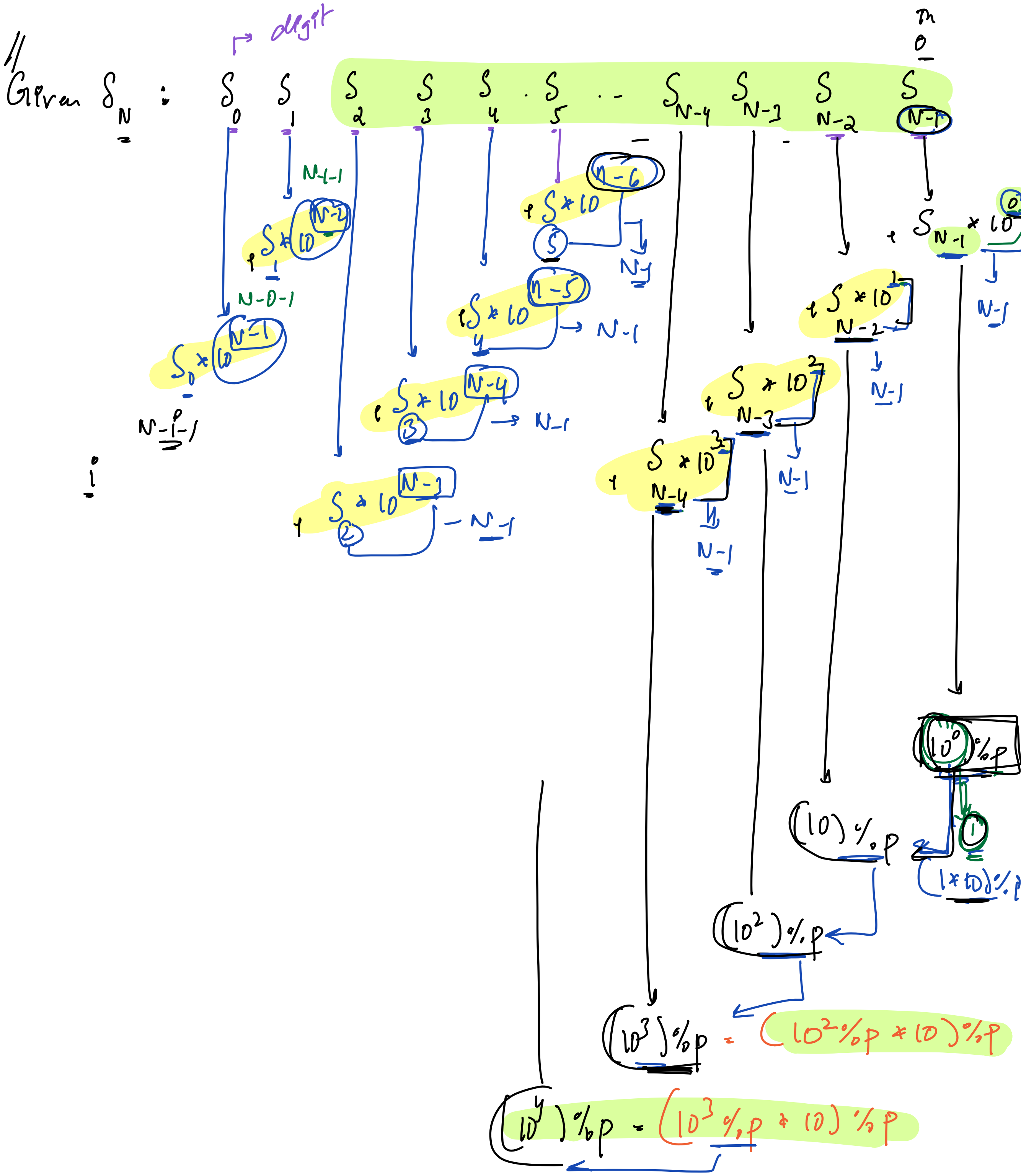
(N)

≈ O(N)

{char to int, O(1)}

Implemented by us

TC: $O(N^2)$ SC: $O(1)$



// Given S_N : S_0 S_1 S_2 S_3 S_4 S_5 .. S_{N-4} S_{N-3} S_{N-2} S_{N-1}
digit
$S_0 * 10^{N-1}$
+ $S_1 * 10^{N-2}$
+ $S_2 * 10^{N-3}$
+ $S_3 * 10^{N-4}$
+ $S_4 * 10^{N-5}$
+ $S_5 * 10^{N-6}$
+ $S_{N-4} * 10^3$
+ $S_{N-3} * 10^2$
+ $S_{N-2} * 10^1$
+ $S_{N-1} * 10^0$
$(10^0)\%p$
$(1*10)\%p$
$(10)\%p$
$(10^2)\%p$
$(10^3)\%p = (10^2\%p * 10)\%p$
$(10^4)\%p = (10^3\%p * 10)\%p$

long ans = 0, exp = 1 }

N = S.size();

i = N-1; i >= 0; i--) {

ans = (ans + s[i] * exp) % p

exp = (exp * 10) % p

}

return ans

p

convert to int

TC: O(N)

SC: O(1)

$10^3 \% p = 10^4 \% p$

p = 7

$1000 \% 7 = (10^4 \% p)$

6

Doubts: C/C++/C#/Java vs Python

Remainder = Divident - (Quotient) * Divisor

a

a/b }

{ is always integer }

$5/2 \Rightarrow 2.5$

$7/3 \Rightarrow 2.333$

$10/6 \Rightarrow 1$

$-10/6 \Rightarrow -1.666$

$-20/3 \Rightarrow -6.666$

$-40/7 \Rightarrow -5.$

$-40 \% 7$

$= -40 - (-40/7) * 7$

$= -40 - (-5) * 7$

$= -40 + 35$

$= -5$

In python ⇒ $5/2 \Rightarrow 2.5$ } Quotient Cannot be float

$8/3 \Rightarrow 2.666$ } ↓ floor

Remainder = Dividend − Quotient * Divisor

↳ floor(a/b)

		floor
$a=10,\ b=4$	$a/b = 2.5$	2
$a=13,\ b=6$	$a/b = 2.12..$	2
$a=-14,\ b=4$	$a/b = -3.5..$	−4

$-60 \% 7 =$

$= -60 - \text{floor}(-60/7) * 7$

$-60/7 = -8.56$ → floor -9

$= -60 - (-9) * 7$

$= -60 + 63$

$= 3$

Doubt : n, q, p

⇒ $\{a^N \% p\}$ → overflow → $\%p$ → long

⇒ $n < 0$:

$[n \% p + p] \% p$

0 0 $\underbrace{1 \quad 1}_{2}$ $\cancel{0}$ 1 $\underbrace{1 \quad 1 \quad 1}_{3}$ 0 1 1 1 1 0 1

2+3+1

3   4

3+4+1 = 8

0 0 1 4 1 0 1

0 → Count 1's left
↳ Count 1's right

%p ⇒