Doubt - 2026-01-22 - Bit Manipulation + Pivot Index

Thursday, 22 January 2026 9:21 PM

a = 0 ⇒ 0000

b = 1 ⇒ 0001

⇒ 0001 ⇒ 1

a = 1 ⇒ 0001

b = 1 ⇒ 0001

0010 ⇒ 2

a = 3 ⇒ 0011

d = 5 ⇒ 0101

1000 ⇒ 8

0011

^ 0101

sum $\boxed{0110}$

0011

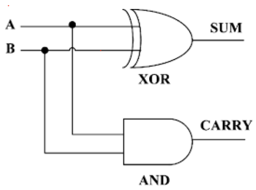
& 0101

Cn 0001

0110 + 0010

6 2

+ ADD + XOR



⇒ 0110

0010

Sum ^ 0100

Carry & 0010

0100 0100

0100

0100

^ 0000

& 0100

0000 1000

0000 0000

0000 1000

^ $\boxed{0000\ 1000}$ ⇒ Sum

& $\boxed{0000\ 0000}$ ⇒ 0

(a, b)

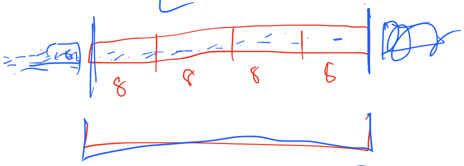
```
while ( b != 0) {
    sum = a ^ b
    carry = a & b
    a = sum
    b = (carry << 1)
}
```

rt

-int_max

$[-\infty, +\infty]$

+int_max

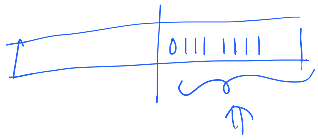


Int

int ans = ∞

byte ⇒ 8 bits ⇒ $\begin{cases} -2^7 \Rightarrow -128 \\ +2^7 - 1 \Rightarrow +127 \end{cases}$

43
32 — 11
8
2
1



a

a = 127

+ 43 ⇒ 0000 0000 0111 1111

— 11 0010 1011

ans = ~~170~~ 0

(-128, +127)

1010 1010

128 32 10

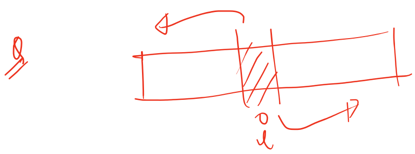
160

170

long ans = 43

int ⇒ +170

0



n-1

for i = 0 .. n-1 {

sum = 0

i=0 [0,] [0 ... n-1]
LSum RSum

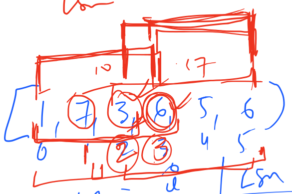
i=1 [0,1] [1 ... n-1]
LSum RSum

i=2 [0,1,2] [2 ... n-1]
LSum RSum

for (j=0; j<=i; j++)
LSum += arr[j]

RSum = 0
for (j=i; j<=n-1; j++)
RSum += arr[j]

if (LSum == RSum)
ret (i)



i=0

i	LSum	RSum	
0	1	28	X
1	8	27	X
2	11	20	X
3	17	17	✓

PreSum

1	8	11	17	22	28
0	1	2	3	4	5

=> O(n) $\Sigma(0, i)$ => O(i) → O(1)

$\Sigma(0, i)$ => Pre[i]

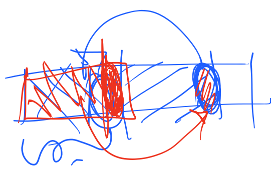
$\Sigma(i, j)$; i≠0 and i<=j => $\Sigma(1, 3)$ => [1] + [2] + [3]

[0] + [1] + [2] + [3]

$\boxed{Pre[j] - Pre[i-1]}$

=> Pre[3] - Pre[0]

$\Sigma(2, 4)$ => [2] + [3] + [4]



for (i=0; i<n; i++)

$\boxed{LSum = Pre[i]}$

$\Sigma(0, i)$

0,0
0 1
0 2
...
(0, n-1)

Pre(i) ✓

// $\Sigma(i, n-1)$
i=0 / i≠0

[0, n-1] => Pre[n-1]

//

if (i==0)

RSm = Pre[n-1]

else

RSm = Pre[n-1] - Pre[i-1]

1, n-1
2, n-1
⋮
(n-1, n-1)

$\Big\}$ Pre[n-1] − Pre[i-1]

Pre

1	8	11	17	22	28
0	1	2	3	4	5

i	LSm (Pre[i])	RSm	
0	1	Pre[n-1] 28	×
1	8	28 − 1 ⇒ 27	×
2	11	28 − 8 = 20	×
3	17	28 − 11 ⇒ 17	✓

```
// prefix[] = new int[n]
prefix[0] = arr[0]
for (i=1; i<n; i++) {
    prefix[i] = prefix[i-1] + arr[i]
}
```

⇒ O(n)

$O(n^2)$ → O(n)

TC ⇒ $\underbrace{O(n)}_{\text{Pre}} + \underbrace{O(n)}_{\text{Cal}}$ ⇒ O(n)

SC ⇒ O(n) ✓

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

RSm = 28

i	LSm	RSm	
0	~~0~~ 1	28	×
1	1 → 8	−1 ⇒ 27	×
2	8 → 11	−7 ⇒ 20	×
3	11 → 17	−3 ⇒ [illegible]	✓

```
Rsm = 0
for (i = 0 ; i < n; i++)   {
        Rsm += arr[i]
}
Lsm = 0
for ( i = 0;  i < n ; i++) {
        Lsm += arr[i]
        if ( Lsm == Rsm) {
            rt (i)
        Rsm -= arr[i]
}
rt (-1)
```

O(n)

O(n)

TC = O(n)

SC = O(1)

$O(n^2)$ ⇒ Brn O(n) ⇒ $\boxed{O(n)}$

O(1)   O(n) $\boxed{O(1)}$