5 questions to be done today

1. Count of set bits
2. Find good days
3. Every element appear thrice except one
4a Subarrays = 1
4b Subarrays = 0
5. Maximum AND Pair .

Q.

int → 32

N = 12

set = 1

1 1 0 0 → 2

N = 21

1 0 1 0 1 → 3

N = 57

1 1 1 0 0 1 → 4

Approach 1 —

int N ·····> 32 bits

↓

4 bytes

N = 12

31 ... 8 7 6 5 4 3 2 1 0

0 ..... 0 0 0 0 0 0 0 0 1 1 0 0

A1

```
count = 0
for ( i = 0 ;   i < 32 ; i++) {
        if ( N & (1<<i)  > 0)
                count ++

}
return count
```

TC
O(32)
O(1)

// K & R Ritchie solution
Kernighan

N = 57 = 111001
& 56 & 111000
111000

N = N & N-1

111001 → 111000 → 110000 → 100000 → 000000

N = 56 = 111000
& N-1 = 55 & 110111
N = N & N-1 110000 ← 48

N = 48 = 110000

& N-1 = 47

& 101111

100000 = 32

N = 32

& N-1 = 31

0

32 = 100000

& 31 = 011111

000000 0

N = 0 ← STOP

A2

```
count = 0
while ( N > 0 ) {
    N = N & N-1
    count ++;
}
return count.
```

TC = O(count of set bits)

57 = 111001

4

A3

```
count = 0
while (N > 0) {
    if (N % 2 != 0)   count ++
    N = N / 2
}
```

div and mod

TC =

$N \longrightarrow \frac{N}{2} \rightarrow \frac{N}{4} \rightarrow \frac{N}{8} \ldots\ldots 1$

$O(\log N)$

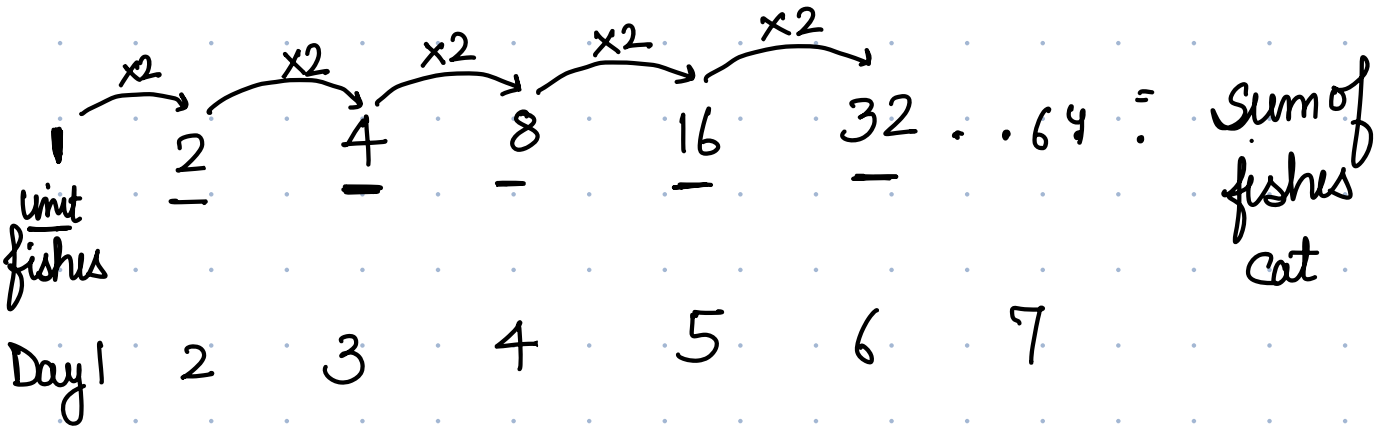
2	57	1
2	28	0
2	14	0
2	7	1
2	3	1
2	1	1
	0	

Q2 Find Good days

$13 = 1\,1\,0\,1$ (positions $2^3\ 2^2\ 2^1\ 2^0$)

Cat

$127 = 1\,1\,1\,1\,1\,1\,1$ (positions $2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$)



5 fishes ⟶ 2 days

16 fishes ⟶ 1 day

21 fishes ⟶ 3 days

$21 = 1\,0\,1\,0\,1$ (positions $2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$)

ans = count set bits of N

Doubts :

int N = 57 → 111001

↓

32 bits

0000...0111001 ✓

..... bitwise operators & | ^

Q0

a	1	2	4	8	16	32 ...
	1	2	3	4	5	

4 → 2 → 1

Q0 = 16 + 4 = 2

Sum of power of 2

Q3 = 16 + 4 + 2 + 1 = 4

Q3 →

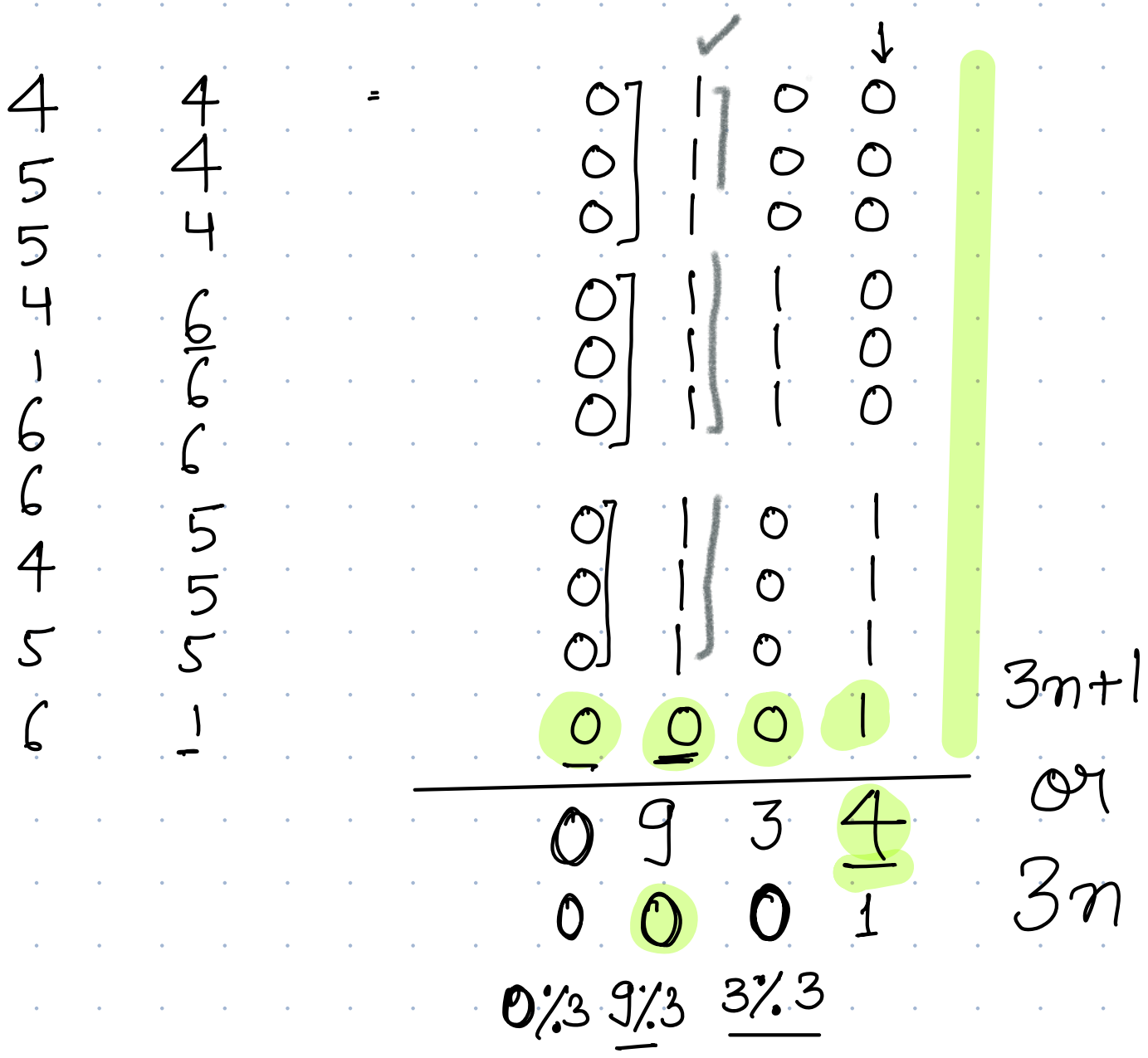
Problem Statement

Given an integer array, all the elements will occur thrice but one. Find the unique element.

Example

Input: [4, 5, 5, 4, 1, 6, 6, 4, 5, 6]
Output: 1

`Only 1 occurs a single time`

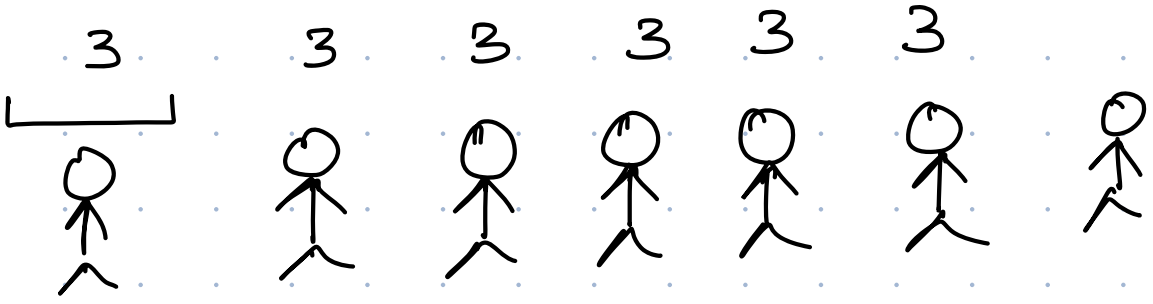


5 + 6 + 7 + 2 = 20

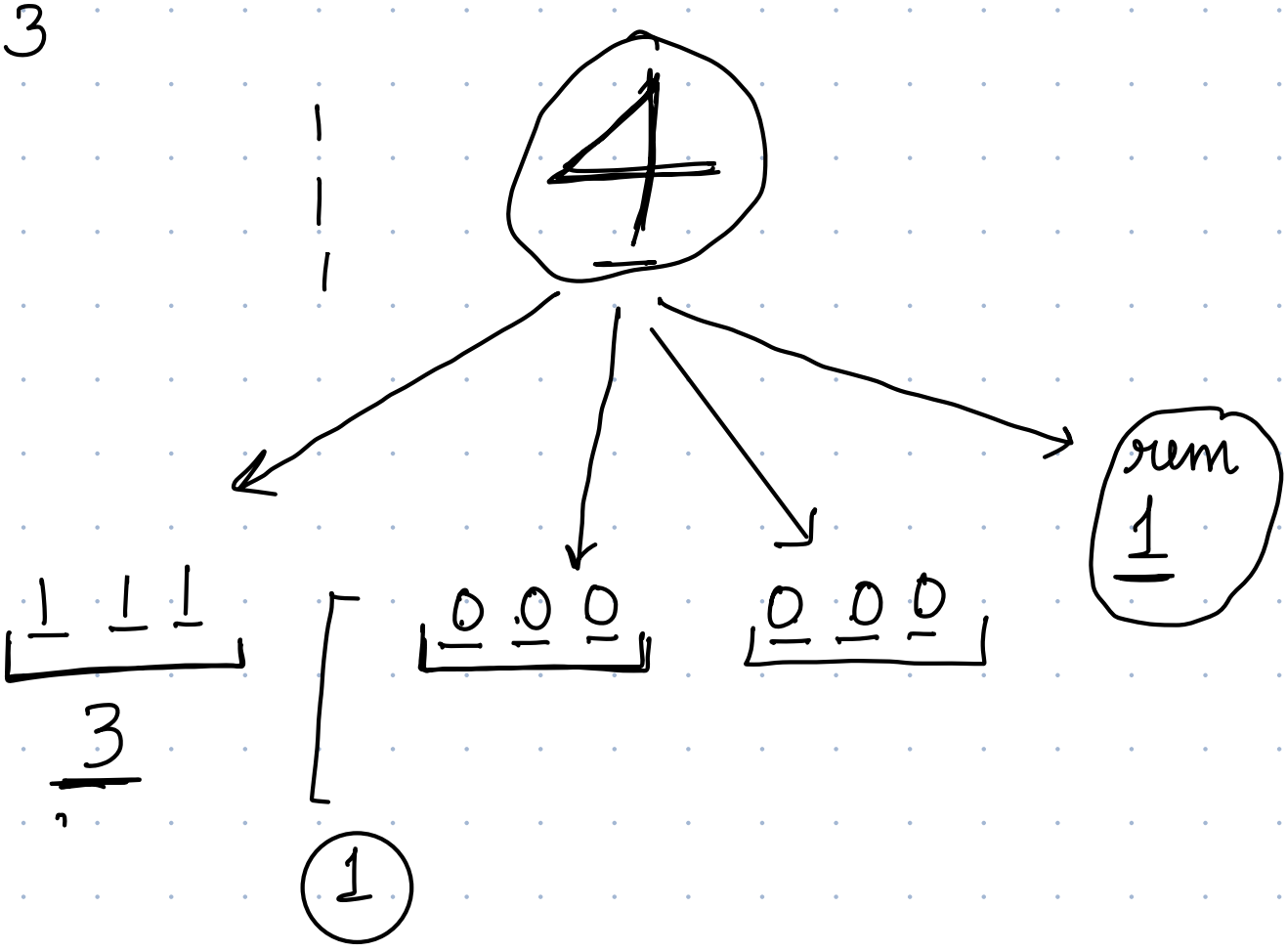
Choco choco choco choco

20/3 = 6

2 rem



3



```
total = 0
for( i=0        i<32        i++) {
        // at pos i , for every no.
    sum = 0
    for(j=0      j<N      j++) {
        if (arr[j] & (1<<i) > 0)
            sum = sum + 1
    }
    rep = sum % 3
    total = total + rep * 2^i
}
return total
```

$3n+1 = 1$

$3n = 0$

0 0 1 1 0 1

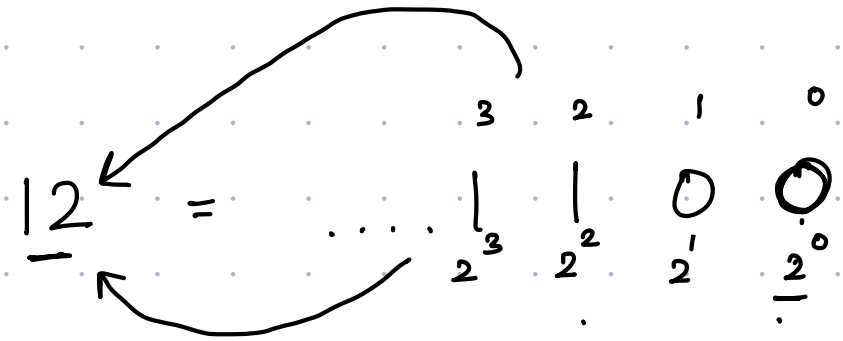
i=31 . . . . i=4 i=3 i=2 i=1 i=0

```
total = 0
for( i=0    i<32    i++) {
    // at pos i , for every no.
    sum = 0
    for(j=0    j<N    j++) {
        if (arr[j] & (1<<i) > 0)
            sum = sum + 1
    }
    3n+1 = 1    rep = sum % 3
    3n = 0      total = total + rep * 2^i  → 1<<i
                                  ↓
                          Math.pow(2, i)
}
return total
```

	31 ...	3	2	1	0
4		0	1	0	0
4		0	1	0	0
4		0	1	0	0
5		0	1	0	1
5		0	1	0	1
5		0	1	0	1
6		0	1	1	0
6		0	1	1	0
6		0	1	1	0
1		0	0	0	1
		0	9	3	4
			sum		
		0	0	0	1
			3x		

$3x \longrightarrow 0$

$3x + 1 \rightarrow 1$



$= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$

$= 8 + 4 + 0 + 0 = 12$

int =

4B

32 bits

31 . . . . 3 2 1 0

Q4

Binary array

arr[i] = 1 or 0

N → [ ]

a. Number of subarrays, whose OR = 0

b. Number of subarrays, whose OR = 1

[1 , 0 , 1]

0 1 2

	OR
1 ---->	1
1 0 ---->	1
1 0 1 ---->	1
0 ---->	0
0 1 ---->	1
1 ---->	1

N = 3

$\frac{N*(N+1)}{2} = 6$

OR = 1 → 5

OR = 0 → 1

Number of subarrays, whose OR = 0 x

Number of subarrays, whose OR = ≥ 1

$$= \text{Total} - x$$

$$\text{Total} = \frac{N*(N+1)}{2}$$

Obs = 2

$$[1, \underbrace{0, 0}, 1]$$

↓

continuous 0's $\longrightarrow$ OR = 0

0 0 1

↑

a single 1] $\longrightarrow$ OR $\rightarrow$ 1

total = 0 + 6 + 6 + 0 + 1

	0	1	2	3	4	5	6	7	8	9
ar[] =	0	0	0	1	0	0	0	1	1	0
	c=1	c=2	c=3	STOP	c=1	c=2	c=3	STOP	STOP	c=1
	6			total = total + c*(c+1)/2, c=0	6			total = total + c*(c+1)/2, c=0	total = total + c*(c+1)/2, c=0	
								0	0	

```
if (ar[i] == 0)
    c++
else if (ar[i] == 1) {
    total = total + c*(c+1)/2
    c = 0
}
```

STOP (i = 9): total = total + c*(c+1)/2, c = 0 → 1 * 2 / 2 → 1

STOP (i = 3): total = total + c*(c+1)/2, c = 0

⇒ STOP: total = total + c*(c+1)/2, c = 0

return total ← No of subarrays with OR = 0

4b

No of subarrays with OR = 1 = Total − No of subarrays with OR = 0

Q5 Maximum AND pair

array of N integers.

arr[i]

arr[] = {5, 4, 6, 8, 5}

5 & 4 = 101 & 110 = 100 = 4

5 & 6 = 101 & 110 = 100 = 4

5 & 8 = 101 & 1000 = 0

6 & 8 = 110 & 1000 = 0000 = 0

6 & 5 = 4

8 & 5 = 0

5 & 5 = 101 / 101 = 5

4 & 6 = 100 / 110 = 4

4 & 8 = 100 / 1000 = 0

4 & 5 = 4

ans = 5 ans

=> N^2 ⟶ 2 loops

=> O(N)

or = [21, 18, 24, 17, 16]

21 = 10101

18 = 10010

24 = 11000

17 = 10001

16 = 10000

4 3 2 1 0

= 1000

max. [a & b] =>

1	0	0	0	1
1	0	0	0	1
4	3	2	1	0

3 where more 1's are aligned

3 → No, 2 → No, 1 → No

at that pos → Count of 1's >= 2

max and pair

Solution

max no of 1's aligneds

app.

Left ⟶ Right

2^4	2^3	2^2	2^1	2^0
1	0	0	0	1
4	3	2	1	0

$2^4 + 0 + 0 + 0 + 2^0 = 17$

count of 1's >= 2 → Y → Align 1

N → 0

L

21	=	1	0	1	0 1
18	=	1	0	0	1 0
24	=	1	1	0	0 0
90	=	0	0	0	0 0
16	=	1	0	0	0 0
		4	3	2	1 0
		1	0	0	0 0

or[i]

0

21 & 18
21 & 24
21 & 16
18 & 24
18 & 16
24 & 16

4

⇒ 16

29	=	1	1	1	0 1
180	=	0	0	0	0 0
24	=	1	1	0	0 0
90	=	0	0	0	0 0
160	=	0	0	0	0 0
		4	3	2	1 0
		1	1	0	0 0

~~29 & 18~~
29 & 24
~~29 & 16~~
18 & 24

~~182 24~~

~~182 16~~ = 24

~~242 16~~

if 29 1. 1 1 0 1

you 18 1. 0 0 1 0

don't 24 1. 1 0 0 0

replace 9 0 1 0 0 1.

16 1 0 0 0 0

4 3 2 1 0

1 1 0 0 1

25

5. [1 0 1

4. [1 0 0

3. 0 0 0

2. 0 0 0

1. 0 0 0

2 1 0

---

1 0 0 → 4

Pseudocode:

for (i = 31 ____ 0) {

= → c=0

for (j=0 j<N j++) {

if (or[j] & (1<<i) > 0) c++

}

if (c >= 2) { make 0 → 0 }

ans → i → 1 }

}

return ans

32 { N

if (c >= 2)
N

}

$$TC = 32(N+N) = 64N$$
$$= O(N)$$