Today's content →

- → A^{th} Magical Number
- → Painters Partition
- → Aggressive Cows.

$$lcm(a,b) = \frac{a * b}{gcd(a,b)}$$

$$[gcd(a,b) * lcm(a,b) = a * b]$$

Q: Find A^{th} magical number.

A number is magical if it is divisible by B or C.

Eg: B = 2, C = 3, A = 8.

1	2 ✓	3 ✓	4 ✓	5 ✗	6 ✓	7 ✗	8 ✓	9 ✓	10 ✓	11 ✗	12 ✓
count:	0	1	2	3		4	5	6	7		8

B.F. Consider all the no's till count $\leq A$ & for every no. we need to check if it is magical or not.

[search space / range for your ans → [1, A * min(B,C)]

target → A^{th} magical no's.]

Q: Given B, C, x. Find count of magical no's from 1 to x.

B = 3, C = 5, x = 35. → (16)

3, 5, 6, 9, 10, 12, 15, 18, 20, 21, 24, 25, 27, 30, 33, 35.

multiples of 3 in [1,35] ⇒ $\frac{35}{3} = 11$

+

multiples of 5 in [1,35] ⇒ $\frac{35}{5} = 7$.

−

multiples of (3*5) in [1,35] ⇒ $\frac{35}{15} = 2$

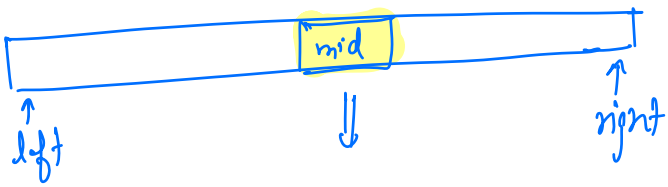
ans = 16

Q: Count of magical nos from [1, 100], B = 9, C = 12.

$$\frac{100}{9} + \frac{100}{12} - \frac{100}{36} = 11 + 8 - 2 = 17$$

(12 → lcm of B, C; 36 → lcm of B, C)

[search space / range for your ans → [1, A * min(B, C)]
target → A^{th} magical no's.]



Count of magical nos ≤ mid.

- < A → Go to right.
- = A → update ans. Go to left.
- > A → Go to left.

tracing.

B=5, C=7, A=3.

search space [1, 15].

target → 3rd magical no.

left	right	mid	Count of magical no's ≤ mid	
1	15	8	$\frac{8}{5} + \frac{8}{7} - \frac{8}{35} = 2$	(Go to right) left = mid+1
9	15	12	$\frac{12}{5} + \frac{12}{7} - \frac{12}{35} = 3$	ans = 12. right = mid-1
9	11	10	$\frac{10}{5} + \frac{10}{7} - \frac{10}{35} = 3$	ans = 10 right = mid-1
9	9	9	$\frac{9}{5} + \frac{9}{7} - \frac{9}{35} = 2$	(Go to right) left = mid+1
10	9		(∵ left > right) break.	

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	0	0	0	1	1	2	2	2	3	3	3	3	4	5

pseudo-code:

```
left = 1 , right = A * min(B,C), ans = 0

while ( left <= right){
        mid = (left + right) / 2;
        count = fun ( B , C , mid); // count of magical no's ≤ mid

        if (count == A){
            ans = mid
            right = mid - 1
        } else if ( count < A){
            left = mid + 1
        } else {
            right = mid - 1
        }
}
return ans;
```

T.C → log (search space)

T.C → O(log(A * min(B,C))

S.C → O(1)

Painter's Partition Problem

Given N boards with length of each board.

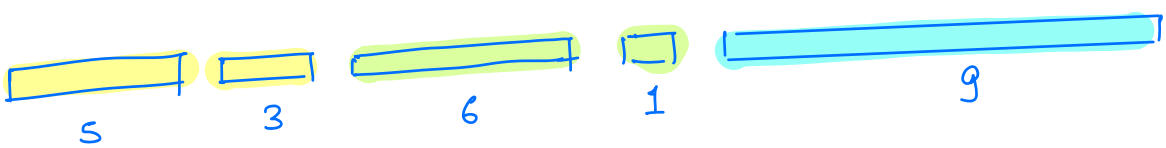
a) A painter takes T unit of time to paint 1 unit of length.
   $l = 3$, time taken $= 3 * T$

b) A board can only be painted by 1 painter.

c) A painter can only paint boards placed next to each other (i.e continuous segment)

Q) Find min no. of painters required to paint all boards in X unit of time. Return -1 if not possible.

N=5.



$\begin{bmatrix} t=2 \\ X=15 \end{bmatrix} \Rightarrow$ {ans = -1} for last board, time taken = 9*2 = 18 units.

$\begin{bmatrix} t=2 \\ X=30 \end{bmatrix} \Rightarrow$ (10 + 6 + 12 + 2)$_{P1}$, (18)$_{P2}$ {ans = 2}

$\begin{bmatrix} t=2 \\ X=20 \end{bmatrix} \Rightarrow$ (10 + 6)$_{P1}$, (12 + 2)$_{P2}$, (18)$_{P3}$ {ans = 3}

pseudo-code:

count = 1 , timeleft = X

for (i = 0 ; i < N ; i++){

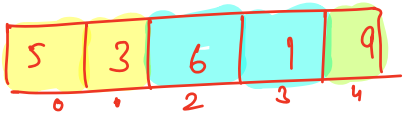
if (A[i] * t > X) {return -1 }

if (A[i] * t <= timeleft) { timeleft -= A[i]*t }

else { count++ , timeleft = X - A[i]*t }

}

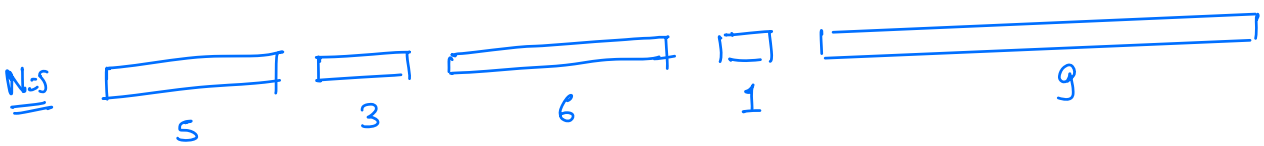
{T.C → O(N)
S.C → O(1)}

Eg. 

count = ~~1~~ 2 3 , X = 20 , t = 2

timeleft = ~~8~~ ~~6~~ 2.

Q) Find minimum time to paint all boards if P painters are available.



{t=2, P=1} (total length * t) = 24*2 = 48.

{t=2, P=2}

max [(5*2) , (3+6+1+9) * 2] = 38

max [(5+3)*2 , (6+1+9)*2] = 32

max [(5+3+6)*2 , (1+9)*2] = 28 → answer

max [(5+3+6+1)*2 , 9*2] = 30

{ min-time → Max element * t , max-time = $\sum A[i] * t$
(1)

target → Minimum time with P painters. }

[B.S on time.]

[no. of Painters $\propto \frac{1}{\text{time to paint}}$]

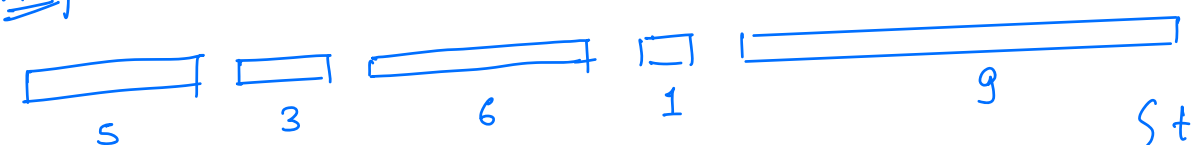
Min no. of painters required with time = mid.

< P → Go to left [right = mid-1]

= P → [update ans.] Go to left [right = mid-1]

> P → { Go to right. left = mid+1 }

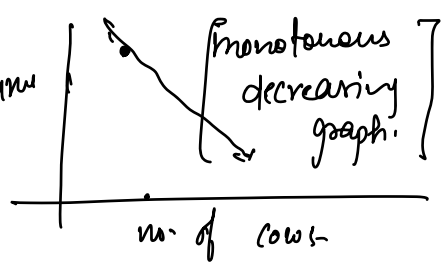
tracing.



{ t = 2. P = 2 }

left	right.	mid.	min Painters required with t = mid.	
18	48	33	2	ans = 33, right = mid-1
18	32	25	3	left = mid+1
26	32	29	2	ans = 29, right = mid-1
26	28	27	3	left = mid+1
28	28	28	2	ans = 28, right = mid-1

B.S. ⇐



Q) Farmer has build a barn with N stalls.

A[i] → location of i^{th} stall in sorted order.

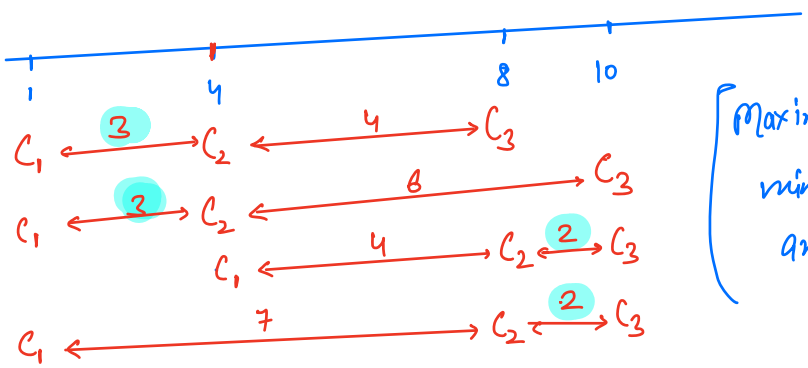
M → no. of cows the farmer has. $\boxed{2 \le M \le N}$

Cows are aggressive towards each other. So, farmer wants to maximise the minimum distance b/w any pair of cows.

Find max possible min distance.

[distance b/w cows $\propto \frac{1}{\text{no. of cows}}$]

A[] → [1 4 8 10] , M = 3



[Maximum possible min distance b/w any pair = 3]

B.F. → $\boxed{{}^{N}C_{M} * M}$ ⇒ Back-tracking.

[1 2 4 8 9], M = 3.

D = 1. $C_1 \xleftarrow{1} C_2 \xleftarrow{2} C_3$

D = 2. $C_1 \xleftarrow{3} C_2 \xleftarrow{4} C_3$

D = 3. $C_1 \xleftarrow{3} C_2 \xleftarrow{4} C_3$

D = 4 $C_1 \xleftarrow{7} C_2$ ✗

D = 5 ✗

D = 6 ✓
D = 7 ✗
D = 8 ✗

target → Maximum value of D (minimum distance)

search space → { 1 ⟶ arr[N-1] - arr[0] }

If it is possible to place M no. of cows with min dist = mid.

Yes → update ans, Go to right

No → Go to left.

Tracing.

[2 (0) 6 (1) 11 (2) 14 (3) 19 (4) 25 (5) 30 (6) 39 (7) 43 (8)] , m = 4.

left.	right.	mid.	Can we place m cows with mid dist ?	
1	41	21	No	right = mid - 1
1	20	10	YES	ans = 10, left = mid + 1
11	20	15	NO	right = mid - 1
11	14	12	YES	ans = 12, left = mid + 1
13	14	13	No	right = mid - 1
13	12			

```
l = 1 , r = arr[N-1] - arr[0]

while( l <= r) {
        mid = (l+r) / 2;
        if (check (arr, mid, M) ) {        → O(N)
            ans = mid
            l = mid+1
        }
        else {
            r = mid-1
        }
}
return ans;
```

T.C → O(N * log(arr[N-1]-arr[0]))
S.C → O(1)

```
boolean check ( arr, dist , M) {
        last_pos = arr[0] , cow = 1 ;
        for( i = 1 ; i < N ; i++) {
            if (arr[i] - last_pos >= dist) {
                cow++
                last_pos = arr[i]
                if ( cow == M) { return true }
            }
        }
        return false;
}
```