Today's content →

- → Painters Partition
- → Aggressive Cows.
- → A^{th} Magical Number (if time permits)

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 $\Rightarrow 3T$

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.

{ 5, 3, 6, 1, 9 }

$\begin{bmatrix} T=2 \\ X=15 \end{bmatrix}$ ans = -1, for last board, time taken = $9*2 = 18$ min

$\begin{bmatrix} T=2 \\ X=30 \end{bmatrix}$ $\underbrace{10 + 6 + 12 + 2}_{P1}$ $\underbrace{18}_{P2}$ ans = 2

$\begin{bmatrix} T=2 \\ X=20 \end{bmatrix}$ $\underbrace{10 + 6}_{P1}$ $\underbrace{12 + 2}_{P2}$ $\underbrace{18 \text{ min}}_{P3}$

[5, 6, 1, 9, 5]

t = 2
x = 20

$\underbrace{10}_{P1}$ | $\underbrace{12+2}_{P2}$ | $\underbrace{18}_{P3}$ | $\underbrace{10}_{P4}$ $\boxed{P = 4}$

code →

```
count = 1 , long timeleft = X

for( i = 0; i < N; i++) {
    if (arr[i] * T > X) {return -1}

    if (arr[i] * T <= timeLeft) {
        timeLeft -= arr[i] * T ;
    }
    else{
        count++;
        timeleft = X - arr[i] * T ;
    }
}
```

$T.C \rightarrow O(N)$
$S.C \rightarrow O(1)$

5, 3, 6, 1, 9.

$T = 2$
$x = 20$

count = ~~1~~
2

timeleft = ~~20~~
~~10~~ 4
8

Painter's Partition Problem - II [Google]

Given N boards with length of each board.

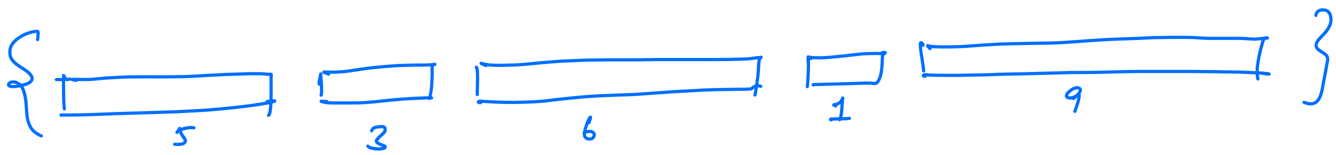
a) A painter takes T unit of time to paint 1 unit of length.

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

N = 5.



$\begin{bmatrix} T=2 \\ P=1 \end{bmatrix}$

$10 + 6 + 12 + 2 + 18 = \underline{48 \text{ min}}$

$\sum arr[i] * T = 24*2$
$= \underline{48 \text{ min}}$

$\begin{bmatrix} T=2 \\ P=2 \end{bmatrix}$

$\max[5*2, (3+6+1+9)*2] \rightarrow 38$ min

$\max[(5+3)*2, (6+1+9)*2] \rightarrow 32$ min

$\max[(5+3+6)*2, (1+9)*2] \rightarrow 28$ min

$\max[(5+3+6+1)*2, 9*2] \rightarrow 30$ min

ans ⇒ 28 min

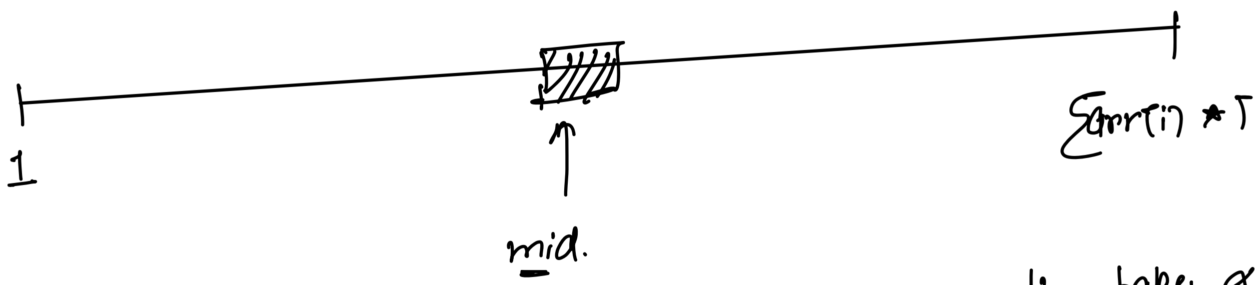
min possible time = 1 $(\text{max-length} * T)$ ✓

max possible time = $\sum arr[i] * T$

search-space.

target → min time to paint all boards with P painters.

BS on time



time taken $\propto \frac{1}{\text{no. of painters}}$

For time = mid, find min no. of painters required to paint all boards

$\leq P$

ans = mid

r = mid-1

$> P$

//Go to right

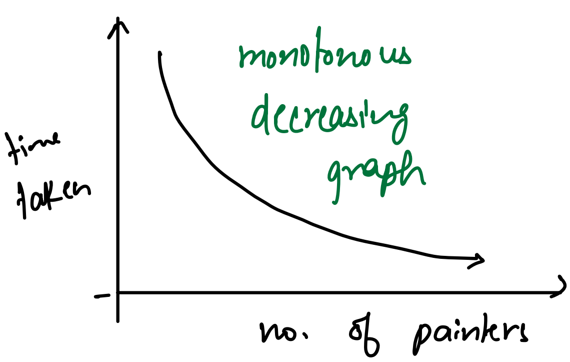
l = mid+1

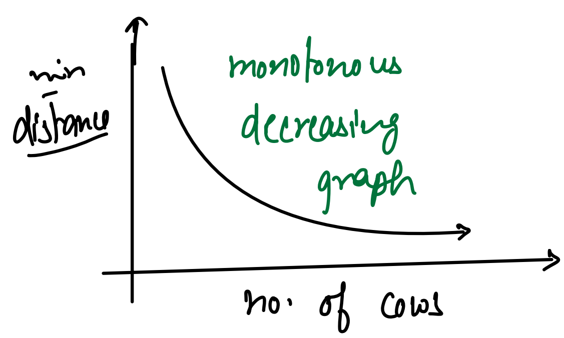
N=5.

{ 5, 3, 6, 1, 9 }

T=2
P=2

l	r	mid	no. of painters required with time = mid ?		
1	48	$\frac{1+48}{2} = 24$	3	→ Go right →	l = mid + 1
25	48	$\frac{25+48}{2} = 36$	2	ans = 36	r = mid - 1.
25	35	$\frac{25+35}{2} = 30$	2	ans = 30	r = mid - 1.
25	29	$\frac{25+29}{2} = 27$	3	l = mid + 1	
28	29	$\frac{28+29}{2} = 28$	2	ans = 28	r = mid - 1
28	27	→ search space exhausted (stop)			





code :→

long l = 1, long r = sum of all elements * T

ans = -1 ;

$1 \le N \le 10^5$

$1 \le T \le 10^6$

$1 \le arr(i) \le 10^6$

```
while( l <= r) {
        long mid = (l+r) / 2 ;

        x = min_painters ( arr[], T, mid);

        if( x == -1){
                l = mid + 1 ;
        }
        else if ( x <= p) {
                ans = mid ;
                r = mid-1 ;
        }
        else {
                l = mid + 1 ;
        }
}
return ans ;
```

T.C → $O(N * \log_2 \Sigma arr[i] * T)$

S.C → $O(1)$

Q) Farmer has build a barn with N stalls.

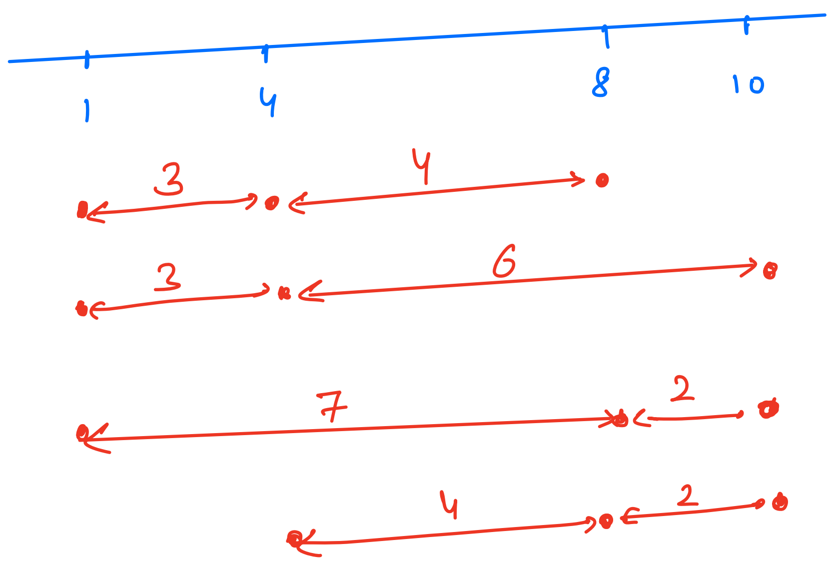
$A[i] \rightarrow$ location of i^{th} stall in sorted order.

$M \rightarrow$ 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.

$A[\,] \rightarrow [1 \quad 4 \quad 8 \quad 10]$, $M = 3$ $\left[\text{distance} \propto \dfrac{1}{\text{no. of cows}} \right]$



$] \Rightarrow 3$

$] \Rightarrow 3$

$] \Rightarrow 2$

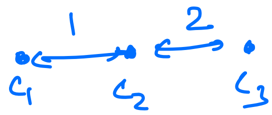
$] \Rightarrow 2$

max → 3

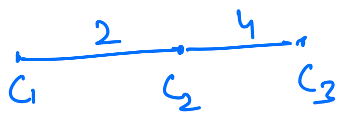
ans = 3.

idea → Consider all combinations [select m stalls out of N stalls] ⇒ Backtracking arr → [1 2 4 8 9]   m = 3.

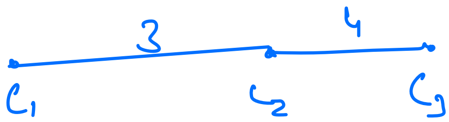
D = 1.



D = 2.



D = 3



D = 4.

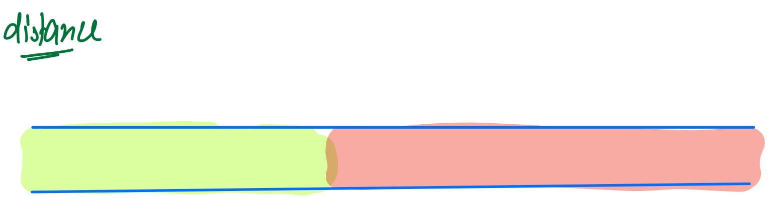
D = 5   Not possible .

D = 6

D = 7

D = 8

D = 9



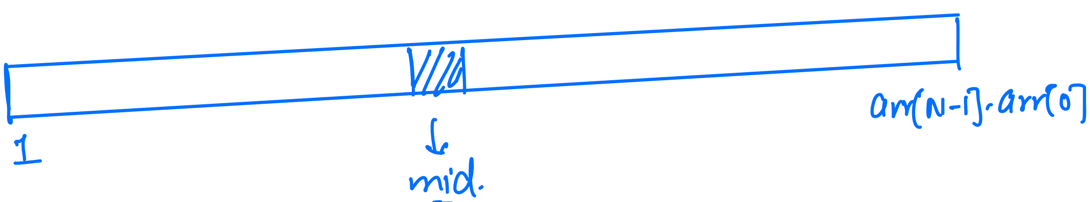
B.S on distance

min-distance (l) → 1

max-distance (r) → arr[N-1] - arr[0]

search-space

target → max value of min-distance (D) possible.



if it is possible to place m cows with min-distance b/w all pairs of cows as mid ?

Yes → ans = mid, l = mid + 1

No → r = mid - 1

arr → [2 6 11 14 19 25 30 39 43] , m = 4

2	6	11	14	19	25	30	39	43
0	1	2	3	4	5	6	7	8

l	r	mid	Can we place m cows with distance = mid ?	
1	41	$\frac{1+41}{2} = 21$	No	r = mid-1
1	20	$\frac{1+20}{2} = 10$	Yes	ans = 10 l = mid+1
11	20	$\frac{11+20}{2} = 15$	No	r = mid-1
11	14	$\frac{11+14}{2} = 12$	Yes	ans = 12 l = mid+1
13	14	$\frac{13+14}{2} = 13$	No	r = mid-1
13	12	→ Stop & return ans.		

pseudo-code.

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

while( l <= r){
        mid = (l+r)/2;

        if ( check( arr, mid, m) == true)){
                ans = mid ;
                l = mid+1 ;
        }
        else{
                r = mid-1 ;
        }
}
return ans;
```

T.C -> O(N log(r))
S.C -> O(1)

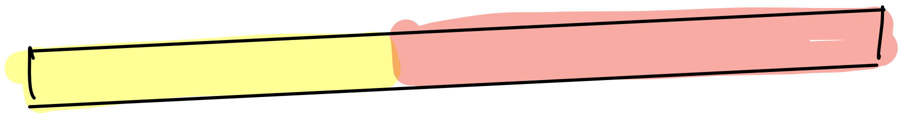
```
boolean check( int[] arr, int dist, int m){
        cows = 1 ,   last-pos = arr[0] ;   N -> arr.length

        for( i = 1 ;  i < N ;  i++){
                if ( arr[i] - last-pos >= dist){
                        cows++ ;
                        last-pos = arr[i]
                        if ( cows == m){
                                return true;
                        }
                }
        }
        return false;
}
```

-> O(N)

Where you can think of applying B.S ?

① - Array is sorted & searching is required

② - 

③ - monotonous increasing / decreasing graph.

④ - log N

→ Complete all assignment problem

→ <u>Revision</u>. Prime No → Sieve, Combinatorics, modular arithmetic

Recursion → Dry-run is very important

Sorting → M.S, I.P, R.P, Merge 2 sorted arrays

& also all sorting algos.

Live Contest → failed → <u>Revise</u>. → Give re-attempt.