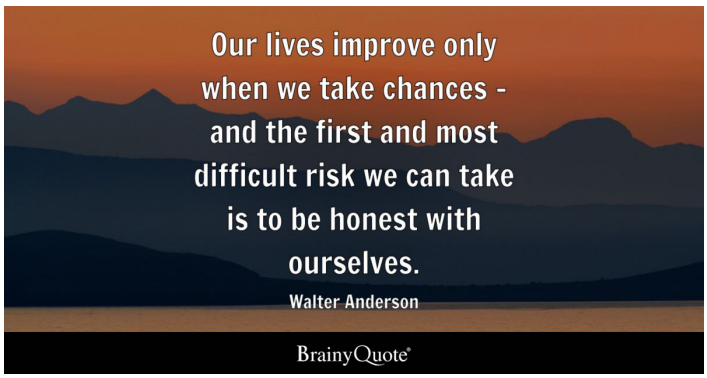
Binary Search 3



Good Evening

Content

01. Painter's Partition I
02. Painter's Partition II
03. Aggressive cows

02. **Painter's partition problem - I**

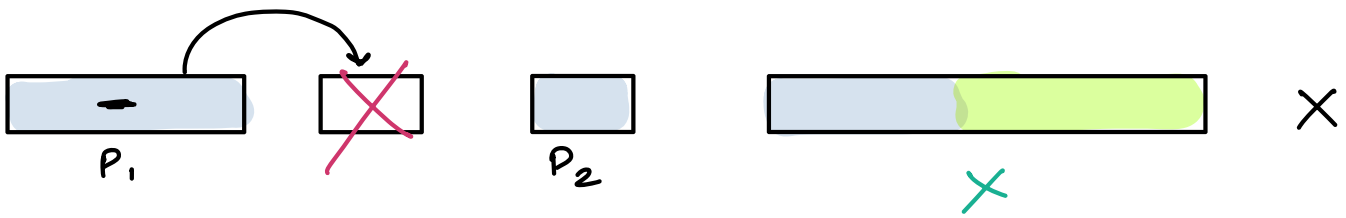
Given N boards with length of each board

(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 continous segment)

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



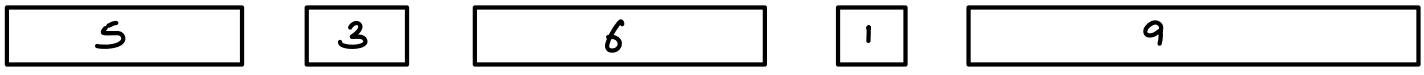
board length = 3

t = 2

Total Time to paint = 3 * t

= 3 * 2 = 6

N = 5



t = 2

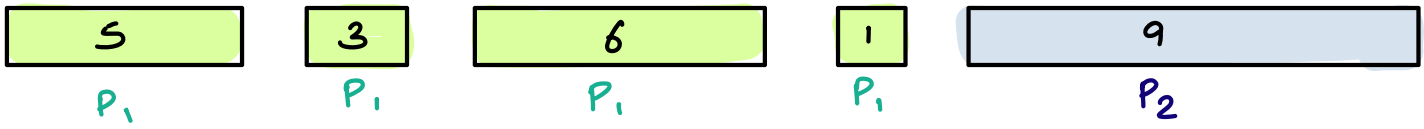
X = 15

since board length = 9 & t = 2

A Painter atleast need 18 mins to paint this board which is > permissible time

Ans = -1

N = 5



t = 2

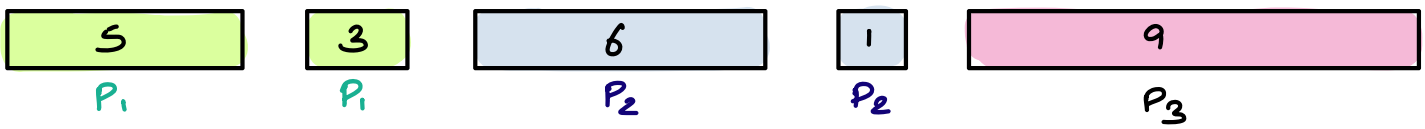
x = 30

$P_1 = 10 + 6 + 12 + 2$

$P_2 = 18$

} 2 painter

N = 5



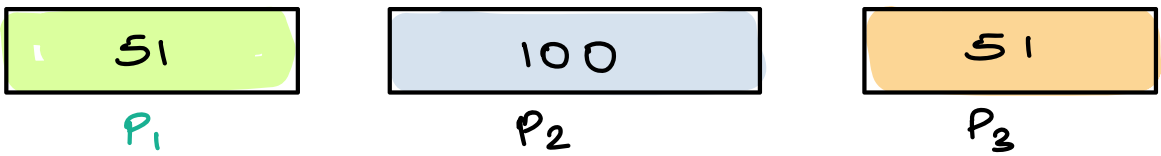
t = 2 mins for 1 length of board

x = 20 mins for 1 painter

$P_1 = 10 + 6$

$P_2 = 12 + 2$

$P_3 = 18$



t = 1

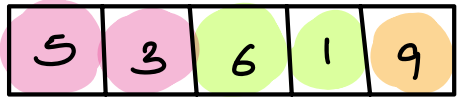
x = 101

Ans = 3 painter

```
minpainter ( arr, x, t)
{
    painter = 1 ,  time left = x

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

        if (ar[i] * t > x) return -1;

        if (ar[i]*t ≤ timeleft){
            timeleft = timeleft - A[i] * t :
        }
        else {
            painter ++:
            timeleft = x - A[i] * t :
        }

    }
}
```

TC: O(n)

SC: O(1)



t = 2

x = 20

P = 2 timeleft = 20

P_1 : timeleft = 20 − 10 = 10

P_1 timeleft = 10 − 6 = 4

P_2 timeleft = 20 − 12 = 8

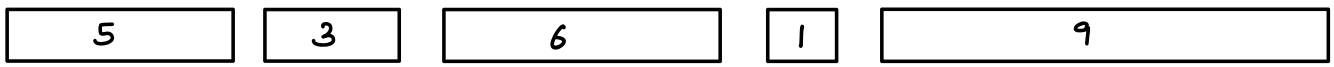
P_2 timeleft = 8 − 2 = 6

P_3 timeleft = 20 - 18 = 2

Painter's partition - II

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

N=5



T = 2 mins to paint 1 unit length

P = 1 painter

Ans = 5 * 2 + 3 * 2 + 6 * 2 + 1 * 2 + 9 * 2

= (total length of boards) * t = 48 mins

T = 2

P = 2

$P_1 = 1 = 5 * 2 = 10$ mins

$P_2 = 4 = 19 * 2 = 38$ min

} Time taken = 38 min

$P_1 = 2 = 8 * 2 = 16$ mins

$P_2 = 3 = 16 * 2 = 32$ mins

} Time taken = 32 min

$P_1 = 3 = 14 * 2 = 28$ mins

$P_2 = 2 = 10 * 2 = 20$ mins

} Time taken = 28 min

$P_1 = 4 = 15 * 2 = 30$ mins

$P_2 = 1 = 9 * 2 = 18$ mins

} Time taken = 30 mins

Eg:-

1

t = 1

P = 1

Ans = 1

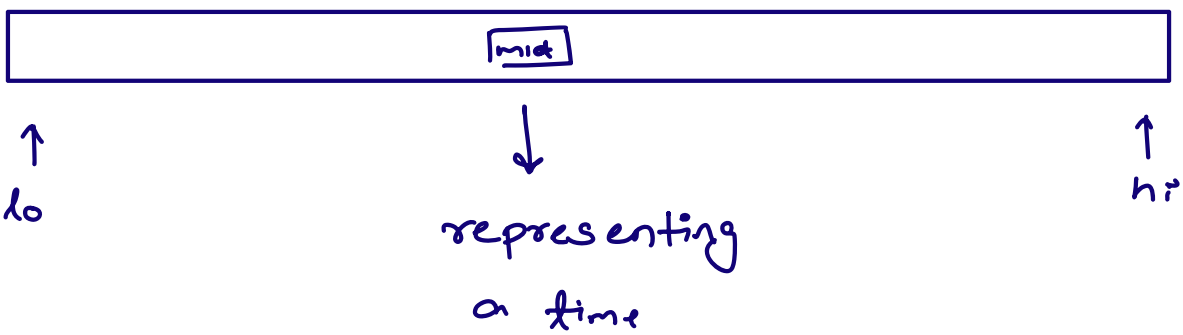
Maximum time = when we have only 1 painter

$$= \sum_{i=0}^{n-1} A[i] * t$$

minimum time = 1 or **maxele * t**

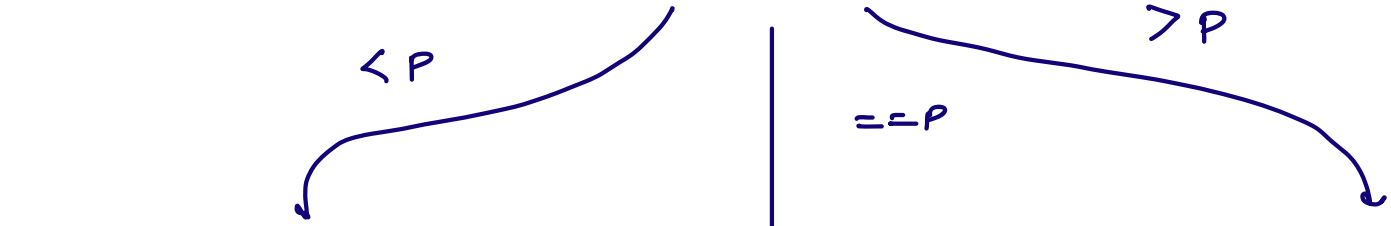
Search space = [maxele * t $\Sigma A[i] * t$]

Target = min time with given P painters



P = no. of painter
given to us in ques

For this mid value, min no.
of painters req to paint the boards



Increase the no. of painters (Decrease time)

go to left side

hi = mid − 1

update ans

hi = mid − 1

Decrease the no. of painters (Increase the time)

go to RHS

lo = mid + 1

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

N = 5 t = 2 P = 2

5	3	6	1	9

lo	hi	mid	min painter req at time = mid	
18	48	33	2	ans = 33 min hi = mid − 1
18	32	25	3	lo = mid + 1
26	32	29	2	ans = 29 hi = mid − 1
26	28	27	3	lo = mid + 1
28	28	28	2	ans = 28 hi = mid − 1
28	27	→ Stop		

$$TC = O(\log(\text{searchspace}) * n)$$

$$= O(\log(\Sigma A[i] * t) * n)$$

$$SC = O(1)$$

Pseudocode → {TODO}

10:17 pm → 10:27 pm

Aggressive cows

Farmer has build a barrier with N stalls

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

M → no. of cows the farmer has. $2 \leq M \leq N$

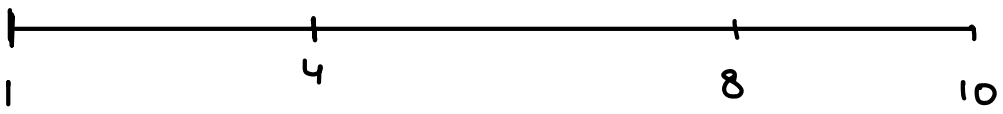
Cows are aggressive towards each other so, farmer wants to maximise the minimum distance b/w any pair of cows. Find max possible min distance

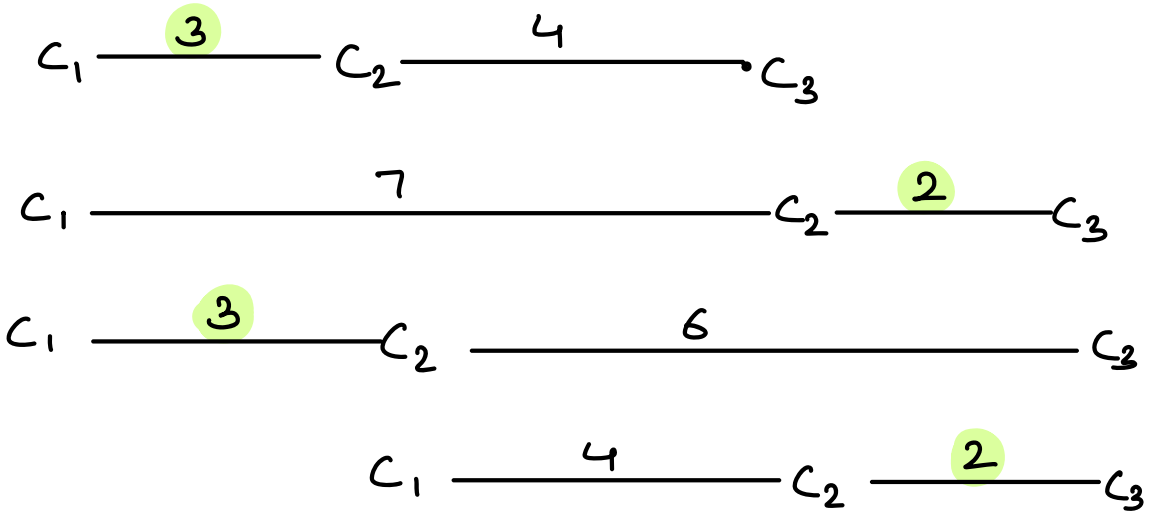
N = 4 ↓

A = {1, 4, 8, 10}

M ↓

cows = 3



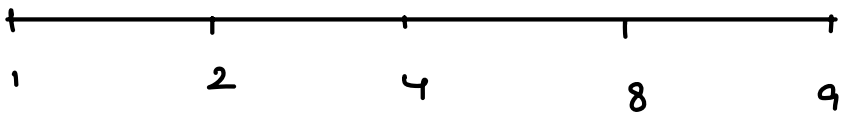


Max possible min dist b/w two rows = 3

Brute force → Generate all the combination where rows can be placed & you need to find min distance & among all min distances, need to find max

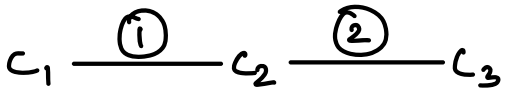
TC : $^{n}C_{m} * m$

Min distance away

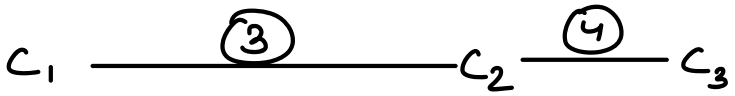


cows = 3

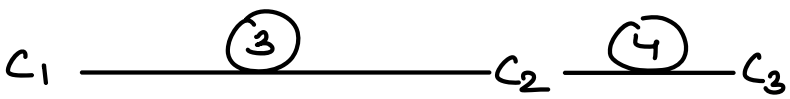
D = 1



D = 2



D = 3



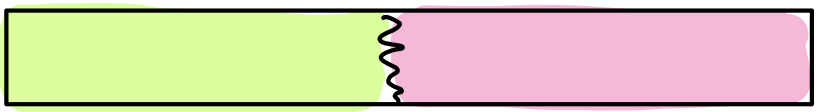
D = 4 Placing the 3 cows in such a manner that they are 4 m away is not possible

D = 5 Not possible

D = 6 ,,

D = 7 ,,

Ans = 3



↑

need to find last green value

Target → Maximum value of D (minimum possible distance)

Search space → [1 A[n-1] - A[0]]

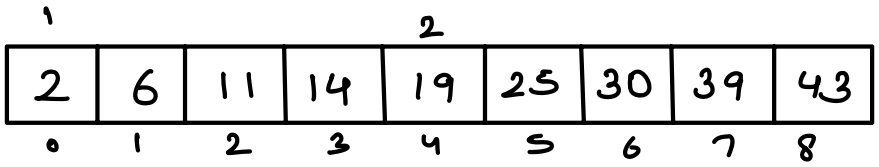
[1 end - start]

For mid distance, if it is possible to place m cows atleast mid distance apart

Yes → Update ans

lo = mid + 1

No → hi = mid - 1



m = 4

lo	hi	mid	Can we place 4 cows atleast mid distance apart
1	41	21	No, hi = mid − 1
1	20	10	Yes, ans = 10 m lo = mid + 1
11	20	15	No, hi = mid − 1
11	14	12	Yes, ans = 12 m lo = mid + 1
13	14	13	No, hi = mid − 1
13	12	→	Stop

Ans = 12 m

* lo = 1

hi = A[n-1] - A[0]

```
while (lo ≤ hi)
    mid = lo + (hi-lo)/2

    if ( check (arr, mid, m) == true) {
        ans = mid;
        lo = mid+1
    }
    else {
        hi = mid - 1
    }
```

TC = O(log (Searchspace) * n)

SC = O(1)

```
boolean check ( arr, dist , m )
    cows = 1      last pos = ar[0]
    for (i=1 ; i<n ; i++ )
        if ( ar[i] - lastpos ≥ dist ) {
            cows++;
            last pos = ar[i];
            if (cows == m) return true;
        }
    return false;
```