Binary Search 3

AGENDA:

→ Painters Partition

→ Agressive cows

Rajat Sharma
Vishal Mosa
Saurabh Ruikar
Rajendra
Akansh Nirmal
Vasanth
PREETHAM
sharath r
Pankaj
Sneha L
suyash gupta
Rajeeb Kumar Nath
Bhaveshkumar
Vimal Kumar
Prajwal Khobragade
Gowtham
Gagan Kumar S
Sanket Giri
Shradha Srivastava
Abhishek Sharma

Current

↓

64 % → 70 %

Painters Partition Problem.

Pay Pay

Q) We have to paint n boards

A_1 A_2 A_3 ..... A_n

There are k painters available and each takes 1 unit of time to paint 1 unit of board.

Find the minimum time to get the job done.

NOTE

→ 1 painter will paint only continuous sections of the board

→ Two painters can't share a board.

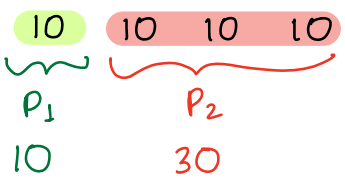
Eg : A = { 10 10 10 10 }

K = 2

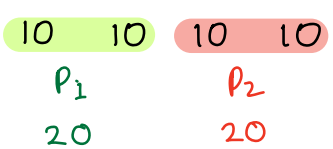
P_1

P_2

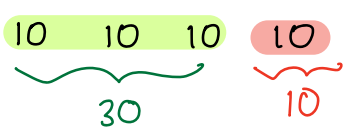
overall time



30



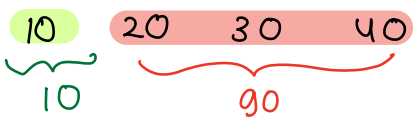
20 any



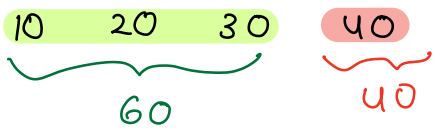
30

Eg $A = \{10 \quad 20 \quad 30 \quad 40\}$

$k = 2$



90



60 ans.

---

will total length of board / total painters = ?

$A = [1\ 2\ 3\ 4 \quad 100]$ $\quad k = 2$

10 100

overall time

100

$A = \quad 2 \quad 5 \quad 3 \quad 8 \qquad k = 4$

min possible answer = max(A)

$A = \quad 2 \quad 5 \quad 3 \quad 8 \qquad k = 1$

max possible answer = sum(A)

⟶ Search space

left = max(A)

right = sum(A)

$k = 4$

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

Assume we want to complete painting in Assume 20 units of time. How many painters do I need?

∵ It will take 5 painters to do the job we can't paint within 20 units of time with 4 painters

$k = 4$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	mid
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6	21
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6	22
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6	20
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6	16
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6	24
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6	40

left	right	mid
10	71	40
10	39	24
10	23	16
17	23	20
21	23	22
21	21	21
22	21	

go	left	ans = 40
go	left	= 24
go	right	———
go	right	———
go	left	= 22
go	right	

Pseudocode

```
// Find min. painters needed to paint all boards
// within given time limit
int paintersNeeded( A[], timeLimit) {
    painter = 1
    time = 0

    for ( i ⟶ 0 to N-1) {
        time += A[i]
        if ( time > timeLimit ) {
            painter ++
            time = A[i]
        }
    }

    return painters
}
```

TC : O(N)
SC : O(1)

20 time limit

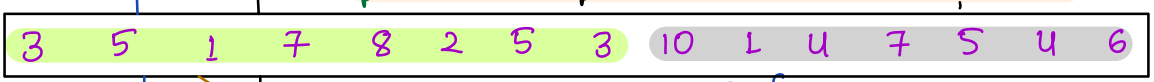
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

painter = ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

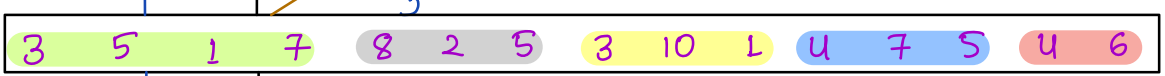
```
int painterPartition ( A[], K ) {
    left  =  max (A)
    right  =  sum (A)
    ans = -1

    while ( left <= right ) {
        mid  =  left + (right - left)/2
        // mid = timeLimit
        painters = paintersNeeded ( A , mid)
        if ( painter <= k ) {
            // store ans and go left
            ans = mid
            right = mid - 1
        }
        else {
            left = mid + 1
        }
    }
    return ans
}
```

O(N)



mid
40
k=4



mid
16
k=4

TC : O(N * log (sum(A) − max(A)))

Break : 22 :23

left = ~~0~~ max(A)

right = ~~N~~ sum(A)

while (left <= right)

. . . . . .

N ⟶ N/2 ⟶ N/4 1

log N

log (~~N~~ − ~~0~~) sum(A) / max(A)

10 ——— 20

10 ⟶ range gets halved.

Linear search solution

```
for timeLimit ⟶ max(A) to sum(A) {
    if ( paintersNeeded ( A, timeLimit) <= K ) {
        return timeLimit
    }
}
```

K = 4

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

first occurrence

Time Limit	10	11	12	13	20	21	22	23	40
painters Needed	9				5	5	4	4	2

whenever there is monotonicity BS can be applied

Aggressive Cows

Given N cows and M stalls, all M stalls are on x axis at different locations, place all N cows in such a way that min dist b/w any 2 cows is maximized.

NOTE → In a stall only one cow can be present
→ All cows have to be placed.

cows = 3

Eg :

0	1	2	3	4
1	2	4	8	9

min dist

1

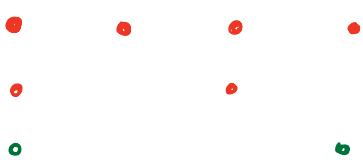
3 any

1

Eg :

cows = 4

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



min dist

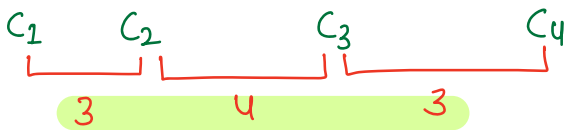
3

8

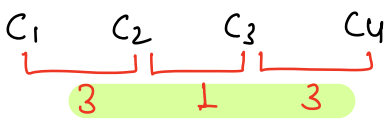
12

any

A = 0 3 4 7 9 10 K = 4



any = 3



1

cows = 4

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

Assume min distance as 20. How many cows can be placed?

→ Always place the first cow at A[0]
{Given that A is sorted}

Only two cows can be placed if min distance b/w any two cows is 20

⇓

decrease min distance so that more cows can be placed

cows = 4

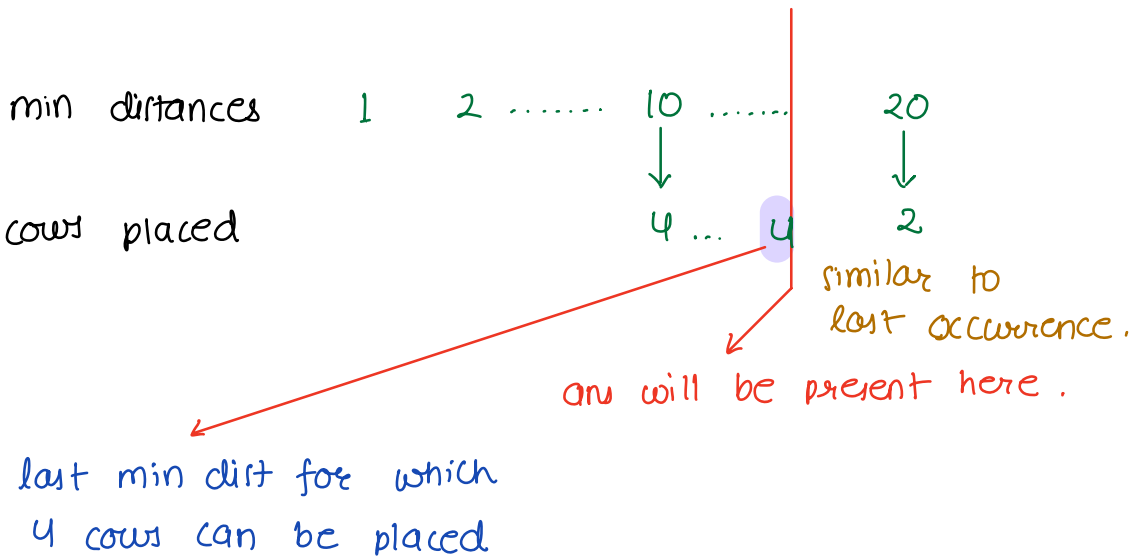
0	1	2	3	4	5	6	7	8
2	6	11	14	19	25	30	39	43

Assume min distance as 10. How many cows can be placed ?

Yes we can place 4 cows at a min dist 10 b/w any two cows.

Store 10 in ans

and go right to maximize min dist.



Search space

min distance b/w any two cows

$\min(A_i - A_{i-1}) \ \forall i$

max distance b/w any two cows

$A[n-1] - A[0]$

target ⟶ max { min dist b/w any two cows }

Pseudo code

```
int moooooo( A [], C ) {
    A.sort()
    left = 1    or
    right = A[n-1] - A[0]
    ans = -1

    while ( left <= right) {
        mid = left + (right-left)/2
        // mid ⟶ min dist b/w any 2 cows
        cows = mooo counter ( A, mid )

        if ( cows >= C ) {
            ans = mid
            left = mid +1
        }
        else {
            right = mid -1
        }
    }
    return ans
}
```

sorting

TC: O(N * log (A[N-1] - A[0]) + N log N)

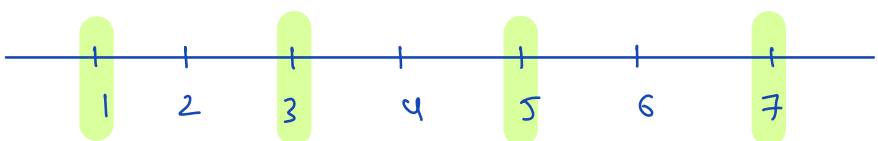
range

```
int moooCounter(A [], mdist) {                    TC: O(N)
    cows = 1
    lastX = A[0]

    for i ⟶ 1 to N-1 {
        if (A[i] - lastX >= mdist) {
            cows ++
            lastX = A[i]
        }
    }
    return cows
}
```

---

cows >= C why ?



Place 2 cows at a min dist of 2

∵ moooCounter can place 4 cows at a min dist of 2
2 cows can also be placed.

cows = 4

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

HW ⟶ Dry Run

★ left right mid

How to identify if a problem can be binary searched on answer ?

⟶ maximize the minimum
⟶ minimize the maximum
⟶ maximize
⟶ minimize

} Think of Binary Search.

Painters partition problem

minimize the maximum

Aggresive cows

maximize the minimum

Contest

⟶ Hashing , Sorting , Searching

8th March 9:00 pm