

Searching - 3

B.S on answer space
 ↳ Square Root
 ↳ A^{th} Magical Number

} Easy - Borden

July - 2021 ↳
 ↳ Goldmn ↳

Question: Aggressive cows
 $(2 \leq k \leq n)$

→ n stalls placed on x -axis ✓
 ($k \geq 2$ and $k \leq n$) ✓

→ k cows

→ place these k cows in n stalls

→ 1 stall → atmost 1 cow

→ cows in stalls such that they are as far as possible

TASIC: Plate



$N = 5, k = 3$

stalls = 0 1 2 4 6 8 10 ✓ Min-dist

$c_1 c_2 c_3$

c_1 c_2 c_3

5 4




Maximise distance between any 2 cows

$N = 9, k = 4$

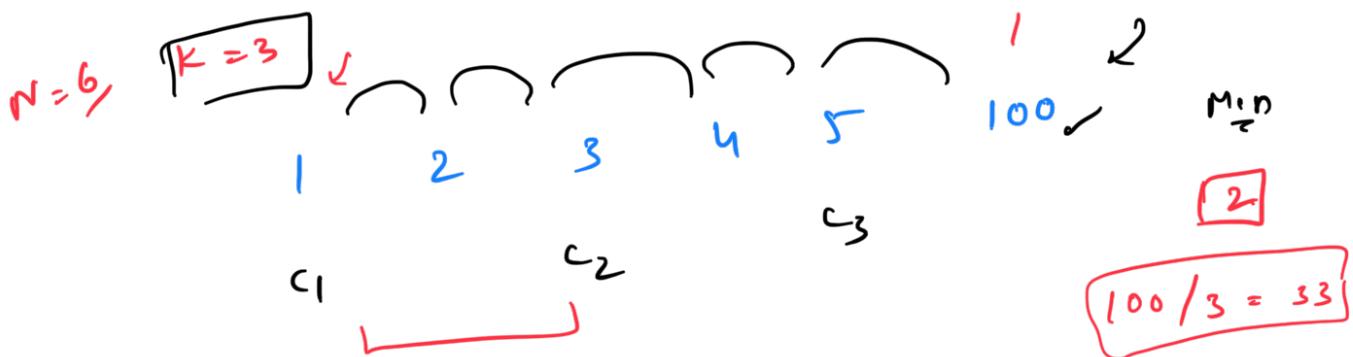
stalls = 0 1 2 3 4 5 6 7 8 9 10 ✓ Min-dist

$c_1 c_2 c_3 c_4$

3



$$13, 14, 15 \dots - \quad (100-1)/k-1 =$$



Brute Force

$N = 10^5$, $k = 2$

all the Possibilities \Rightarrow $\boxed{k=3}$
 $\text{prev} = \beta^{11}$

Toy stalls $\left[\begin{matrix} -1 & c_1 \\ 1 & 3 \end{matrix} \right] \rightarrow \left[\begin{matrix} -1 & c_1 & c_2 \\ 1 & 3 & 7 \end{matrix} \right] \rightarrow \left[\begin{matrix} -1 & c_1 & c_2 & c_3 \\ 1 & 3 & 7 & 11 \end{matrix} \right] \rightarrow \dots \rightarrow \left[\begin{matrix} -1 & c_1 & c_2 & c_3 \\ 1 & 3 & 7 & 11 & \dots & 30 \end{matrix} \right]$

$\boxed{\text{Backtracking}} = \boxed{\frac{N!}{(N-k)! k!}}$ Combinations

T.C: $\binom{N_c}{k} \propto O(N)$

Binary Search

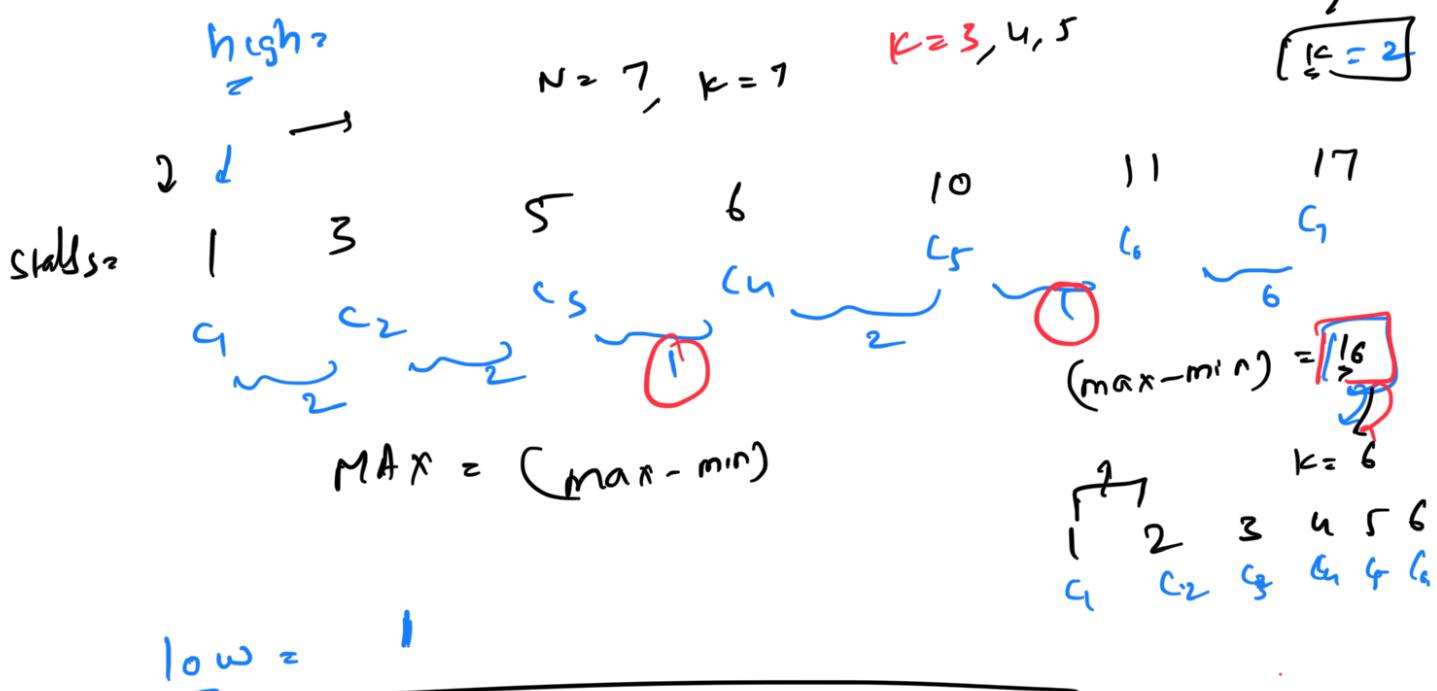
$\Rightarrow \text{Maximum} / \text{Minimum} \Rightarrow \text{Binary}$

↓
Search space

↓
lowest possible

↓ highest possible

$[1 \dots n-1]$



Search Space:

low = 1

high = stalls[n-1] - stalls[0]

Simpler Problem:

check if we can place k cows where any 2 cows are at least D distance apart

$k=4, D = 8 \text{ units}$

5 cows

positions = 2 6 11 14 19 25 30 39 43
 $c_1 c_2 c_3 c_4 c_5$

Given -

Can we place k cows such that every 2 units apart?
 can \rightarrow place k cows are at least 7 units apart?

$K = 4, D = 15$

2	6	11	14	19	25	30	39	45
---	---	----	----	----	----	----	----	----

Stalls = $16, 17, 18, \dots$

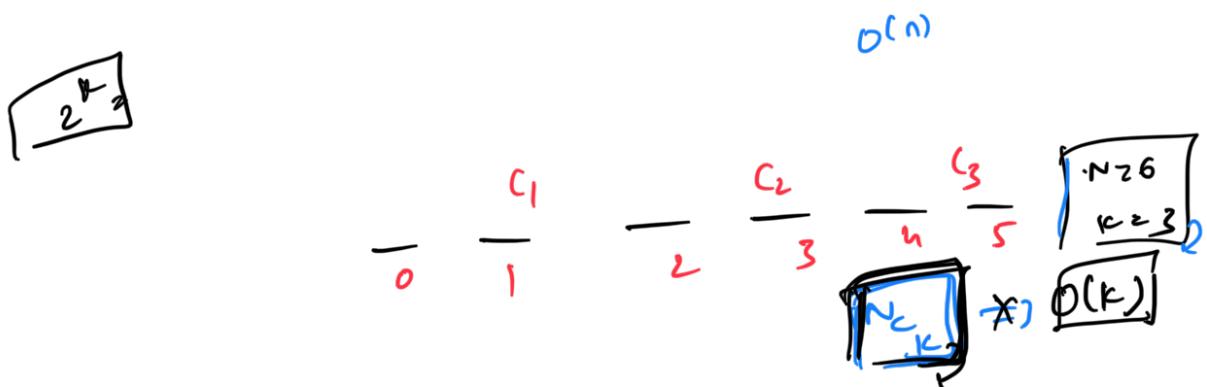
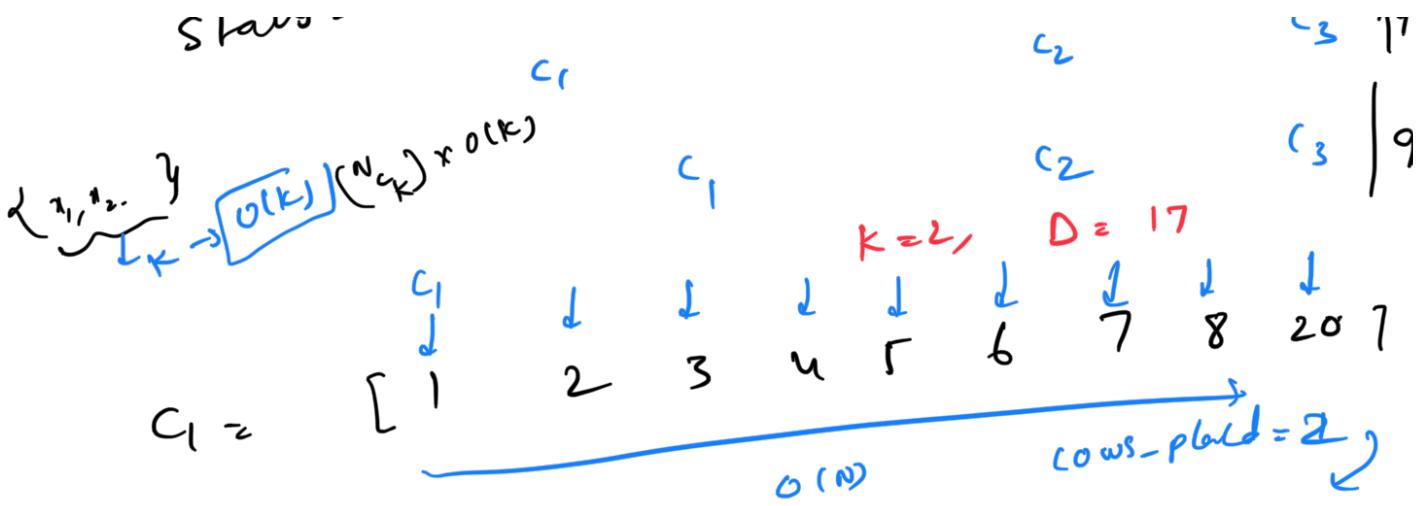
bool check(stalls[], N, K, D) {
 recent-placed = stalls[0];
 cows-placed = 1;

for ($i = 1$; $i < N$; $i++$) {
 if ($stalls[i] - recent_placed \geq D$) {
 recent-placed = stalls[i];
 cows-placed++;
 if ($cows_placed == k$) return true;
 }
}

if ($cows_placed \geq k$) return true
 else false;

$O(n^2)$

$M = 1, 2, 6, 11, 17, 20$ $|_{\text{K=3}}$

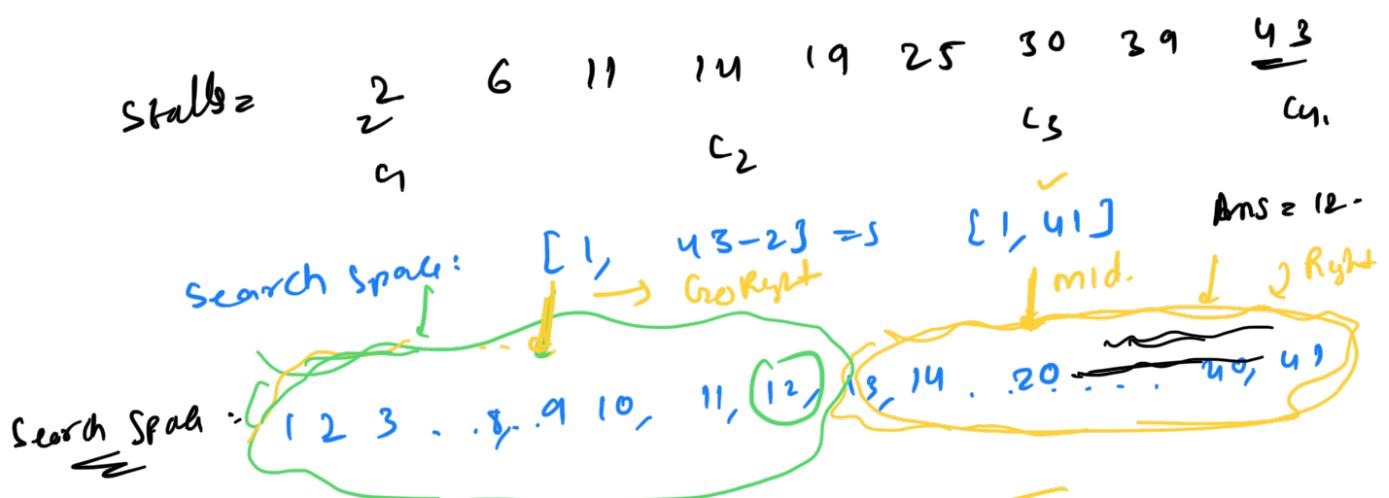


\rightarrow

$\{1, 3, 5\} \Rightarrow O(K)$

(max-min)

$K=4$



ans = 9
if check(mid) == true

ans = mid
low = mid + 1

check(mid) == false
high = mid - 1



Search Space: 

$\text{check}(D)$

$\text{check}(\text{mid}) = \text{true}$

$\text{ans} = \text{mid}$

$\text{low} = \text{mid} + 1$

$\text{check}(\text{mid}) = \text{false}$

$\text{high} = \text{mid} - 1$

int AggressiveLoser(Stalls[], K) {

$\text{low} = 1$
 $\text{high} = \text{stalls}[n-1] - \text{stalls}[0]$

while ($\text{low} \leq \text{high}$) {

$\text{mid} = (\text{high} + \text{low}) / 2;$

if ($\text{check}(\text{mid}) = \text{true}$) {
 $\text{ans} = \text{mid};$
 $\text{low} = \text{mid} + 1;$

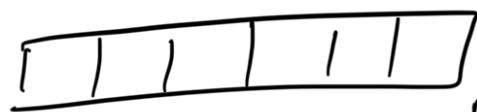
else if ($\text{check}(\text{mid}) = \text{false}$)
 $\text{high} = \text{mid} - 1;$

Hard
on

4

return ans;

Surfing



log n

log n

Search:

$[0, n-1]$

size of Search Space: $[1, \text{stalls}[n-1] - \text{stalls}[0]]$

$\text{size} = O(\text{stalls}[n-1] - \text{stalls}[0])$

iterations in $\approx \log(\text{stalls}[n-1] - \text{stalls}[0]) \times O(n)$

B-S

$\dots - \text{stalls}[0];$

$T.C = O(n \cdot \log(\text{stalls}))$

Question: Painters Partition Problem

there are N boards each with a length
 like have K painters
 A painter can paint only continuous
 boards
 Painting i unit of boards takes t_i ✓
 Find the minimum time to get all the
 boards painted [Not sorted]

$N = 4, K = 2$				max time
B_1 10	B_2 20	B_3 30	B_4 40	
P_1	P_2	P_2	P_2	90
P_1	P_1	P_1	P_2	70
P_1	P_1	P_1	P_2	60

$$P_1: 10^{(S)}$$

$$P_2: 20+30+40$$

$$= 90(S)$$

$$P_1: 30$$

$$P_2: 70$$

$$P_1: 60$$

$$P_2: 40$$

< 60
50, 58, 59,

$$N=4, K=4, t=1$$

10	20	30	40
P_1	P_2	P_3	P_4

Search Space:

high :	$\text{sum}(\text{boards})$
low :	$\text{max}(\text{boards}) \rightarrow$
0, 11	

$\min(b - c)$

$N=4, K=4$

10 20 30 40

$10(S) ?$

$$\begin{array}{cccc}
 p_1 & p_2 & p_3 & p_4 \\
 \downarrow & & & \\
 p_1 : 10 & & & \\
 p_2 : 20 & & & \\
 p_3 : 30 & & & \\
 \boxed{p_4 = 40} & & &
 \end{array}$$

Simpler Question :

Given n boards and k painters, check if we can paint all the boards where each painter paints for almost T time.

$N = n, K = 2, T = 80$

boards =	10 p1	20 p1	30 p1
----------	----------	----------	----------

$$40 \\ p_1$$

$$p_1 = 10 + 20 + 30 \quad \}$$

$$p_2 = 40$$

Can we paint all the boards with 2 painters where every painter paints for almost $90(s)$

$$T = 81, 82, 83, \dots, 89$$

$$N = 4, \quad \boxed{k = 2}, \quad \boxed{T = 50} \quad \text{?}$$

boards =	10 p1	20 p1	30 p2
----------	----------	----------	----------

$$40 \\ p_3$$

✓

$$\text{if } T < 50$$

$$\boxed{T = 49, 48, 47, \dots} \quad \boxed{k = 2}$$

$$p_1 = 10 + 20$$

$$p_2 = 30$$

$$p_3 = 40$$

... with 2 painters who

Can we paint all boards for almost 40(s)
using painter paints for almost 40(s)

$$N=4, K=2$$

Ans = 60

boards = 10 20 30 40

$$\text{low} = 40$$

$$\text{high} = 100$$

mid → Right
40 41 42 ... 58, 59

$\text{check}(\text{mid}) = \text{False}$

$$\text{ans} = \text{mid}$$

$$\text{high} = \text{mid} - 1$$

[40, 100]

60, 61, 62 ... 99, 100

60 61, 62 ... 99, 100

int $\text{paintersPartition}(\text{boards}\}, N, K)$ {

 low = $\max(\text{boards})$;

 high = $\sum(\text{boards})$;

 while ($\text{low} \leq \text{high}$) {

 mid = $(\text{high} + \text{low})/2$;
 if ($\text{check}(\text{mid}) = \text{True}$;
 and $\text{ans} = \text{mid}$;
 high = $\text{mid} - 1$);

 else {
 low = $\text{mid} + 1$;
 }

} return low;

$O(N)$

SS: [max(boards), $\sum(\text{boards})$] -

$O(N \times \log(\sum(\text{boards})) - \max(\text{boards}))$

$$N=4, K=2$$

$$T=70$$

... 20

30

40

02.

cheat

boards =

10
p1

11
p1

p1

p1

low = [40, 100]

check(70) = true
check(54) =

low
→ 40

high

100

mid

(70)

Action

ans = 70 ✓
high, mid - 1. ✓

40

69

54

low = mid + 1.

55

69

62

check(62) = true
ans = 62 ✓
high, mid - 1.

55

61

58

check(58) = false
low, mid + 1.

59

61

60

check(60) = true
ans = 60 ✓
high, mid - 1.

59

59

57

check(57) = false
low, mid + 1.

60

59

59

ans = 60

low = mid + 1.

a, b

$$a \times b = \text{gcd}(a, b) \times \text{lcm}(a, b)$$

Resource

$\text{lcm}(a, b) =$

$$\frac{a \times b}{\text{gcd}(a, b)}$$

$O(\log(n))$

Euclidean

maximum / minimum

$\log(a, b)$

maths: gcd

/

Euc

find max/min,
there will be

binary search
if w the

→

whenever we want to
in most of the case
a solution using
cannot guarantee it.

\rightarrow all most optimal 80%

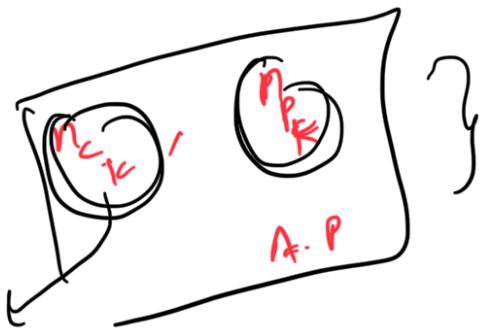
check fast

(low, high)

check (mid) \rightarrow DP / Greedy

Euclidean

Research Papers



max, min \Rightarrow
Greedy, DP

Binary Search

Greedy, DP

$a, b \rightarrow$ Euclidean
Asker

n_{C_k}, n_{P_k}
20 - 40%
Extra Sets

$N=3$

2^N subsets

$A = \{1, 2, 3\}$
 $N = \{1\} \{2\} \{3\} \{1, 2\} \{1, 3\} \{2, 3\} \{\} \{1, 2, 3\}$

8 subsets

$\underbrace{(2)(2)}_{2^2} \cdot 2^3 = 2^5$

$N=5$

1 3 5 10 12 $= 2^5$

Very Very Comp

Recursion / Backtracking

Recursion call stat

$\alpha \rightarrow K \rightarrow$

Iteration

Stack

n_{C_K}, P_{D_K}

30-40

$K = 2$

stalls =

 $N=9$
 $k=3$
 $k=9$
 $k=9$
 $k=2$
 $c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7 \quad c_8 \quad c_9$

 high = $\lceil \text{stalls}[n-1] - \text{stalls}[0] \rceil$
 low : $\lceil \frac{\min \text{ difference between any consecutive elements}}{2} \rceil > (\max - \min)$
 $\text{low} = 2$
 $N=4, \quad k=2, \quad 3$
 A =

$\text{low} = 1$

{ , 2 3 4 5 6]

$$f_{n+1} \omega_3(\dots) \Big] + o(n)$$

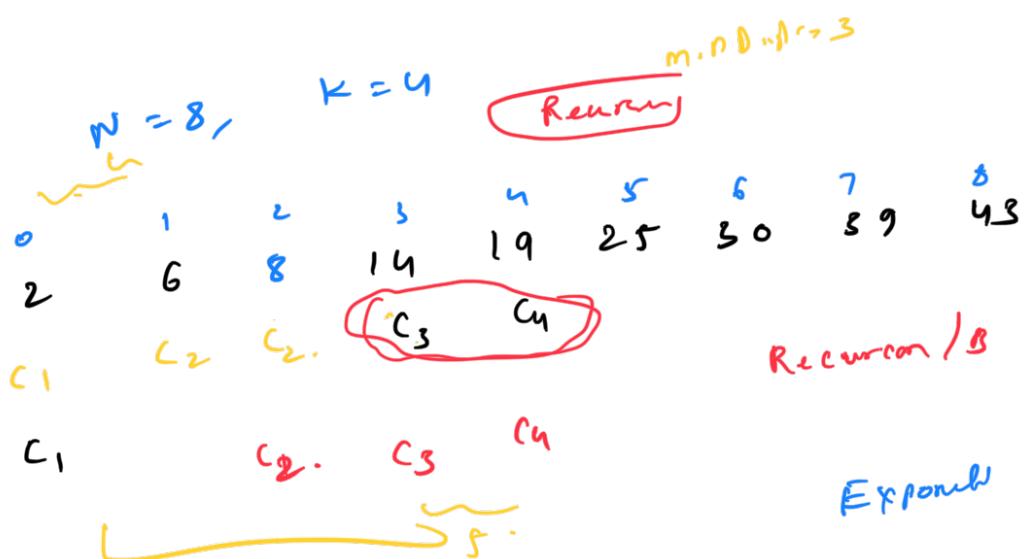
T.C: $O(2^N)$

Recursion:

Can we break a bigger problem into smaller problems

$O(N)$

Stalls =



Question: Aggressive cows

```
bool check(int stalls[], int N, int k, int minDist) {
    int recent_placed = stalls[0];
    cows_placed = 1;

    for(int i=1;i<N;i++) {
        if(stalls[i] - last_placed >= minDist) {
            recent_placed = stalls[i];
            cows_placed++;
        }
    }
    if(cows_placed >= k)
        return True;
    else
        return False;
}
```

```

int AggresiveCows(int stalls[], int n, int k){
    low = 1, high = stalls[n-1]-stalls[0];
    int ans = 0;
    while(low <= high) {
        mid = (high + low) / 2;

        if(check(stalls, n, k, mid) == true) {
            ans = mid;
            low = mid + 1;
        }
        else
            high = mid - 1;
    }
    return ans;
}

```

Question: Painter's Partition problem

```

bool check(int boards[], int n, int k, int maxTime) {
    int painters = 1;
    int time = 0;

    for(int i=0;i<N;i++) {
        time = time + boards[i];

        // Recently added board cross max time
        if(time > maxTime) {
            // Assign it to next painter
            time = boards[i];
            painters++;
        }
    }

    if(painters > k)
        return False;
    return True;
}

```

```
int paintersPartition(int boards[], int n, int k){  
    lo = max(boards), high = sum(boards);  
    int ans;  
  
    while(low <= high){  
        mid = (high + low)/2;  
        if(check(boards, n, k, mid) == true){  
            ans = mid;  
            high = mid - 1;  
        }  
        else  
            low = mid + 1  
    }  
    return ans;  
}
```