Agenda

1. Painter Partitions Problem
2. Aggressive Cows Problem

Revision Quiz 1

Revision Quiz 2

$N = 36$ ans $= \sqrt{36} = 6$

$x \times x = N$

$N = 38$ ans $= \sqrt{38} = 6$

mid * mid < N

Right side

Revision Quiz 3 : TC : $O(\log_2 N)$

Revision Quiz 4 : SC of recursive Binary Search : $O(\log_2 N)$

Q. Given an array representing board length (N boards). Paint all boards if T time is available with you. Find min no. of painters required to get job done.

- 1 unit of length takes 1 unit of time to paint.
- 1 board can't be divided in painters.
- Painters will paint consecutive boards only.

Ex : < 3 5 1 7 6 9 1 5 > T=9

3	5	1	7	6	9	1	5
P_1	P_1	P_1	P_2	P_3	P_4	P_5	P_5

ans = 5

$P_1 \to 9$

$P_2 \to 7$

$P_3 \to 6$

$P_4 \to 9$

$P_5 \to 6$

TC : O(N)

SC : O(1)

```
int minNoOfPainters (arr, T) {
    int cnt = 1
    int work = 0
    for (i=0 ; i<N ; i++) {
        if (work + A[i] > T) {
            cnt++
            work = A[i]
        }
        else {
            work = work + A[i]
        }
    }
    return cnt
}
```

Q. Given an array representing board length (N boards). Paint all boards if T time and P painters are available with you.
Can these painters do the job within T time?

Ex : < 3 5 1 7 6 9 1 5 > T=9, P=4

P1 P1 P1 P2 P3 P4 P5 P5

Min of 5 painters > P ⇒ False

Ex : < 3 5 1 7 6 9 1 5 > T=9, P=6

Min of 5 painters ≤ P ⇒ True

```
boolean isPossibleToPaintBoard(arr, T, P)
    int cnt = 1
    int work = 0
    for (i=0 ; i<N ; i++) {
        if (work + A[i] > T) {
            cnt++
            work = A[i]
        }
        else {
            work = work + A[i]
        }
    }
    if (cnt ≤ P)
        return true
    else
        return false
```

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

1. Given an array representing board length (N boards). Paint all boards if P painters are available with you. Find min. time required to get job done by the painters.

- 1 unit of length takes 1 unit of time to paint.
- 1 board can't be divided in painters.
- Painters will paint consecutive boards only.

Ex : < 3 5 1 7 6 9 1 5 > $P = 1$

ans = $3 + 5 + 1 + 7 + 6 + 9 + 1 + 5$

$= 37$

Ex : < 3 5 1 7 6 9 1 5 > $P = 2$

ans = 21

< 3 | 5 1 7 6 9 1 5 > 34

3 (P_1) 34 (P_2)

< 3 5 | 1 7 6 9 1 5 > 29

8 (P_1) 29 (P_2)

< 3 5 1 7 | 6 9 1 5 > 21

16 (P_1) 21 (P_2)

< 3 5 1 7 6 9 1 5 >   22

22 15

P_1 P_2

Greedy Approach: ✗

Divide total work by total no. of painters

ans = 60

A = [10, 20, 30, 40]   P = 2

$$\frac{\text{Total work}}{\text{Painters}} = \frac{100}{2} = 50$$

A = [10, 20, 30, 40]   P = 2

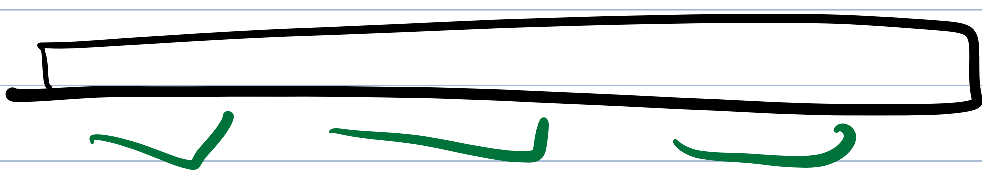
30 70   70

P_1 P_2

A = [10, 20, 30, 40]

60 40   60

P_1 P_2



P_1 P_2 P_3

x y z → max(x, y, z)

A = [1, 2, 3, 4, 100]   P = 2   ans = 100

10 (P_1)   100 (P_2)   100

Approach 2 : Binary Search on Time

Search space : l = max(arr[ ]) | r = sum(arr[ ])

Lower limit time	Upper limit
Best case time to paint all boards	Worst case time to paint all boards
P = N	P = 1

Target : Min time to complete the job

Condition

l   mid   r

Is my mid the answer ?

Can we paint all boards in ≤ mid time (given k painters)

l   mid   r

5 s　　　10 s　　　15 s

Y ↙　　　↘ N

ans → 10 s　　　≤ 10 s ✗
Go to left　　　Go to right

1s	2	3	4	5	6	7	8	9	10	11	....
✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	

N=15　P=3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A[ ] =	3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

$\underbrace{3\ 5\ 1\ 7\ 8\ 2\ 5\ 3}_{P_1}$ $\underbrace{10\ 1\ 4\ 7\ 5\ 4\ 6}_{P_2}$

l	r	mid	Can we paints boards in $\le$ mid time?	Where ans to go?
10	71	40	2 < 3	update ans 40 Go Left
10	39	24	4 > 3	No Go right

N=15　K=3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A[ ] =	3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

$\underbrace{3\ 5\ 1\ 7\ 8}_{P_1}$ $\underbrace{2\ 5\ 3\ 10\ 1}_{P_2}$ $\underbrace{4\ 7\ 5\ 4}_{P_3}$ $\underbrace{6}_{P_4}$

25	39	32	3 = 3	update ans 32 Go Left

N=15　K=3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A[ ] =	3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

P_1 (indices 0–6), P_2 (indices 7–12), P_3 (indices 13–14)

l	r	mid
25	31	28

3 = 3

update ans 28
Go Left

N=15 K=3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A[ ] =	3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

P_1 (indices 0–5), P_2 (indices 6–10), P_3 (indices 11–14)

l	r	mid
25	27	26

3 = 3

update ans 26
Go left

N=15 K=3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A[ ] =	3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

P_1 (indices 0–5), P_2 (indices 6–10), P_3 (indices 11–14)

l	r	mid
25	25	25

3 = 3

update ans 25
Go left

N=15 K=3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A[ ] =	3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

P_1 (indices 0–4), P_2 (indices 5–10), P_3 (indices 11–14)

l	r
25	24

l > r End of BS STOP

ans = 25

```
int minTimeToPaint (arr[], N, P) {

    l = max (arr[])    } Iterate arr[]
    r = sum (arr[])    }
    int ansTime

    while (l <= r) {

        mid = l + (r-l)/2

        // Is mid answer ? Can we paint
        all boards ≤ mid time ?

        bool canPaint = isPossibleToPaintBoard (arr, mid, P)

        if (canPaint == true) {
            ansTime = mid
            r = mid - 1        // left
        }

        else {
            l = mid + 1        // right
        }
    }

    return ansTime
}
```

Formula for TC :

log (range) × TC of feasibility check

log (sum (arr[]) - max (arr[]) × N

× N

O(log (sum (arr[]) - max (arr[]) × N)

10 : 45

https://notability.com/n/2JAcoN5Gd1EuaoPkgcJpAb

Aggressive Cows

2. Given N cows and M stalls, all M stalls are located at different locations on x-axis. Place all cows such that min. distance b/w any 2 cows is maximised.

$N \leq M$

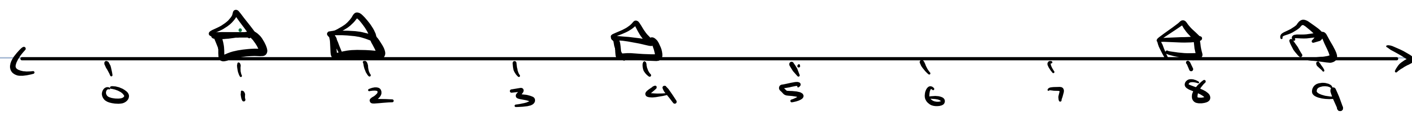
NOTE:

1) Only one cow in a stall
2) Place all the cows

M = 5

Stalls [ ] = [1 2 4 8 9]   N = 3

ans = 3

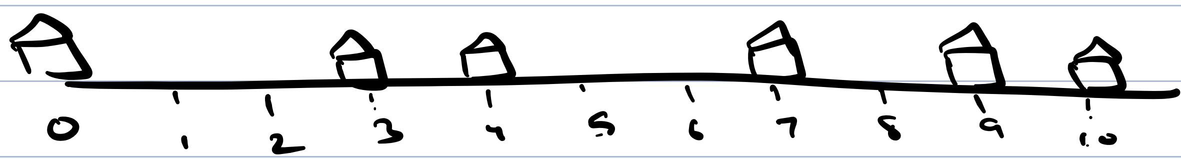


$C_1 \xleftrightarrow[1]{} C_2 \xleftrightarrow[2]{} C_3$ 1

$C_1 \xleftrightarrow[3]{} C_2 \xleftrightarrow[5]{} C_3$ 3

$C_1 \xleftrightarrow[6]{} C_2 \xleftrightarrow[1]{} C_3$ 1

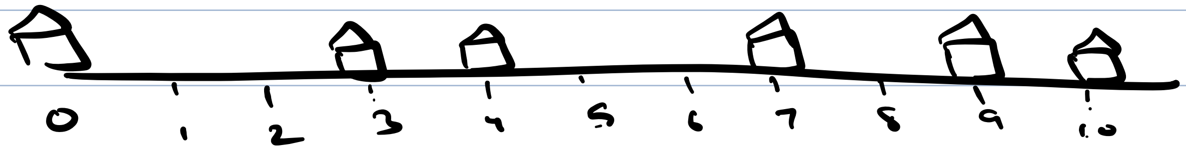
ans = 4



M = 6
N = 3

$C_1 \xleftrightarrow[3]{} C_2 \xleftrightarrow[1]{} C_3$ 1

$C_1 \xleftrightarrow[4]{} C_2 \xleftrightarrow[6]{} C_3$ 4



M=6
N=4

$C_2 \xleftrightarrow{4} C_2 \xleftrightarrow{3} C_3 \xleftrightarrow{3} C_4$ 3

Maximize dist
(shortest dist b/w any pair of cows)

Target : Maximize the distance

Search space :

l = min (arr[i] - arr[i-1])

↓

lower limit

N = M
no. of cows = no. of stalls
so every stall will have 1 cow

r = arr[N-1] - arr[0]

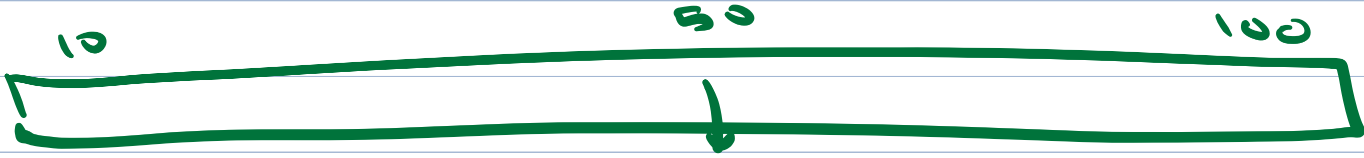
↓

upper bound

N = 2 (2 cows)
Place both of them at extreme stalls

Condition : Can mid be answer ?

Can I place my cows atleast mid distance apart ?



Y ↙ ↘ N

dist ≥ 50 Left

ans = 50

Right

---

stalls
M = 9

cows
N = 4

ans = 12

stalls =	[2	6	11	14	19	25	30	39	43]
22	c_1					c_2			
12	c_1			c_2			c_3		c_4
17	c_1				c_2			c_3	
14	c_1				c_2			c_3	
13	c_1				c_2			c_3	

l	r	mid	Place cows ≥ mid dist apart ?	where to go ?	ans
3	41	22	2 < 4	× Left	
3	21	12	4 = 4	ans = 12 Right	12

13 21 17 3 < 4 × Left

13 16 14 3 < 4 × Left

13 13 13 3 < 4 × Left

13 12 l > r stop Binary search

```
bool isPossiblePlaceCows (arr, M, mid, N) {
    int cnt = 1  // placed at arr[0]
    int lastPlaced = arr[0]
    for (i=1 ; i < M ; i++) {
        if (arr[i] - lastPlaced >= mid) {
            cnt++
            lastPlaced = arr[i]
        }
    }
    if ( cnt >= N )  return true
    else  return false
```

}

```
int maxShortestDist (int arr[], int m, int n)
{
    l = 1 / min (adjacent arr[])
          ↓
      for (i=1 ; i<m ; i++) {
        l = min (l, a[i] - a[i-1])
      }

    r = a[m-1] - a[0]
    int ans
    while (l <= r) {
        mid = l + (r-l)/2
        Is mid my answer? can I
        place all cows atleast mid dist.
        apart?

        boolean canPlace = isPossiblePlaceCows
                           (arr, m, mid, n)

        if (canPlace == true) {
            ans = mid
            l = mid+1        // right
        }
        else {
            r = mid-1        // left
        }
    }
```

}

return ans

}

TC : $O(\log(r - l) \times N)$

$r - l$: a[N-1] - a[0]

l / min of diff of adjacent

SC : $O(1)$

BS on answer

① 2 - 3 parameters and constraints

② Maximize / Minimize on some parameter

BS → Monotonic problems

P = 3
ans = 10

1	2	3	4	5	....	9	10	11	12	13 ...
×	×	×	×	×		×	✓	✓	✓	✓
F	F	F	F	F	...	F	T	T	T	T

1 2 3 4 5 6 7 8 9 10
✓ ✓ ✓ ✓ ✓ ✓ ✗ ✗ ✗ ✗

① Linked List

② Rice Prob

Matrix ques

$A = 4$

1, 4, 5, 6, 7, 8, 10

sum of x ele

$x = A - A/3$