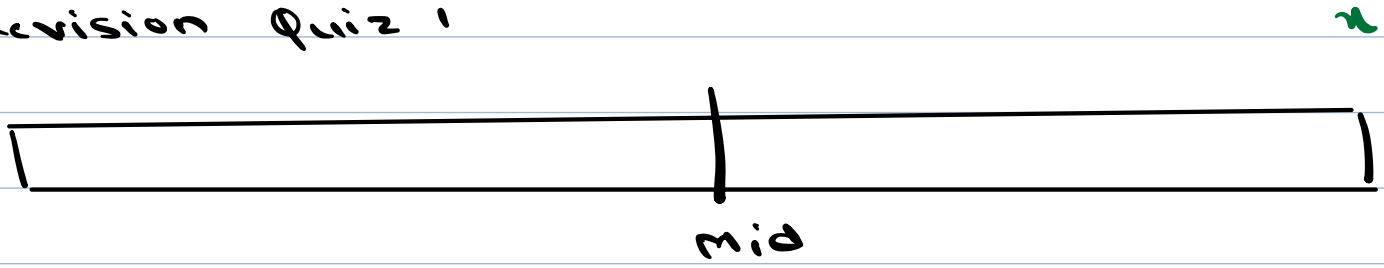
Agenda

1. Painter Partitions Problem
2. Aggressive Cows Problem

Revision Quiz 1



Revision Quiz 2

$x \times x = N$

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

$N = 38$ 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

P_1 P_1 P_1 P_2 P_3 P_4 P_5 P_5

ans = 5

$P_1 \rightarrow 9$ $P_5 \rightarrow 6$

$P_2 \rightarrow 7$

$P_3 \rightarrow 6$

$P_4 \rightarrow 9$

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

P_1 P_1 P_1 P_2 P_3 P_4 P_5 P_5

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)

2. 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 | 34

P_1 P_2

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

8 | 29

P_1 P_2

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

16 | 21

P_1 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

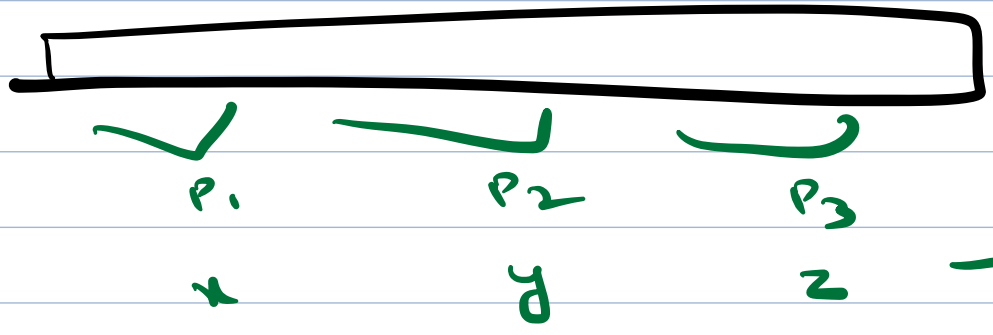
30 | 70 → 70

P_1 | P_2

A = [10, 20, 30, 40]

60 | 40 → 60

P_1 | P_2



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

10 (P1) 100 (P2) 100

Approach 2 : Binary Search on Time

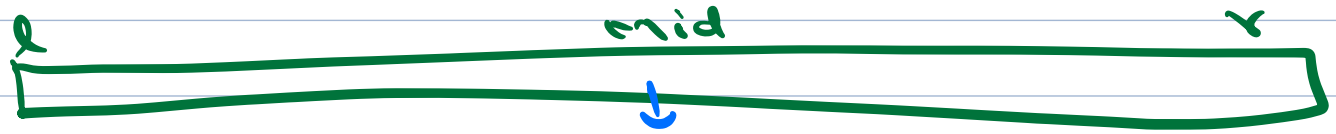
Search space : l = max(arr[]) | r = sum(arr[])

l = Lower limit time
Best case time to paint all boards
P = N

r = Upper limit
Worst case time to paint all boards
P = 1

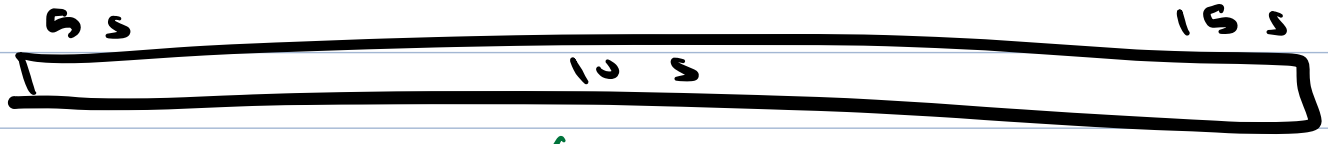
Target : Min time to complete the job

Condition



Is my mid the answer ?

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



Y → ans → 10 s, go to left

N → ≤ 10 s ✗, 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

P_1 P_2

l r mid — Can we paints boards in ≤ 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

P_1 P_2 P_3 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 P_2 P_3

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 P_2 P_3

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 P_2 P_3

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 P_2 P_3

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