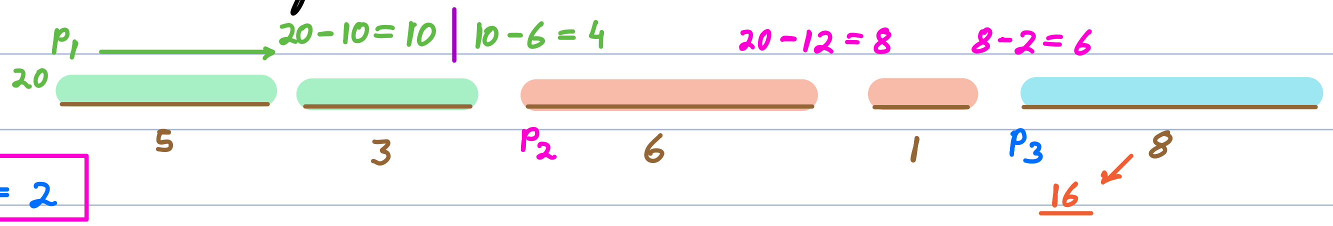
Q→ Painter's Partition Problem

1) Given N boards with lengths in array A.
2) It takes T unit of time to paint 1 unit of board.
3) Each board should be painted by only 1 painter.
4) A painter can only paint consecutive boards.

A → Find min # painters required to paint all boards in X unit of time.



$\boxed{T = 2}$

X = 10 → Ans = -1 (Not Possible)

X = 20 → Ans = 3

A = [10 40 25 10 5]

T = 3   30 120 75 30 15

X = 200   P1 (30 120)   P2 (75 30 15)   Ans = 2

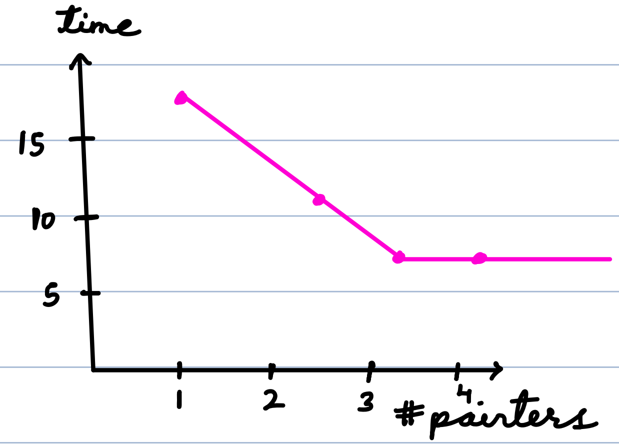
```
cnt = 1        remT = X
for i → 0 to (N-1) {
    c = A[i] * T
    if (c > X)    return -1
    if (c <= remT)    remT -= c
    else {  cnt++
            remT = X - c   // new painter painting current
```

```
    }
}  return cnt
```

TC = O(N)   SC = O(1)

B → Find the min time required to paint all boards if K painters are available. (T = 1)

A = [2 5 8 3]



$$\boxed{\#\text{painters} \propto \frac{1}{\text{min time}}}$$

#painters	Time

1 → 2 + 5 + 8 + 3 = 18

2 → min (max (2, 5 + 8 + 3),
max (2 + 5 , 8 + 3),
max (2+5+8 , 3))
= min (16 , 11 , 15) = 11

3 → min (max (2, 5, 8+3),
max (2, 5 + 8, 3),
max (2+5, 8 , 3))
= min (11, 13, 8) → 8

4 → 8

BS on Answer

```
// Search space
  l = ∀i max (A[i])     // N painters available
  r = Σi A[i]           // 1 painter available

while (l <= r) {
    mid = l + (r-l)/2
    // check if mid is answer
    cnt1 = Painters (A, mid)
    cnt2 = Painters (A, mid-1)
    if ( cnt1 <= K && cnt2 > K)    return mid
```

```
    // Decide left / right
    if (cnt1 > K)     l = mid + 1
    else              r = mid - 1
}
```

$TC = \underline{O(N * \log(\Sigma A[i] - \max(A[i])))}$

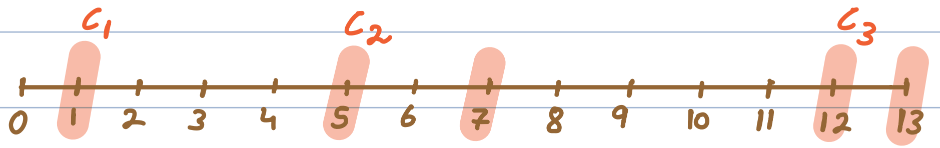
$SC = \underline{O(1)}$

Q → <u>Aggressive Cows</u>

1) Given N stalls & their location in A[] in ascending order.

2) We need to place all cows maximising the min distance b/w any pair of cows.

A → Find the max # cows that can be placed if min dist. b/w any pair should be >= X.



A = [1 5 7 12 13]

X = 3 → Ans = <u>3</u>

C_1 C_2 C_3

A = [2 6 11 14 19 25 30]

X = 10 → Ans = <u>3</u>

```
cnt = 1        last = A[0]
for i → 1 to (N-1) {
    if (A[i] - last >= X) {
```

```
        cnt ++      last = A[i]
    }
}  return cnt
```

TC = O(N)   SC = O(1)

B → Find the max possible min distance b/w any pair of cows if total K cows are to be placed.

c_1 ... c_2

A = [0 3 4 7 9 10]

K = 2 → Ans = 10

$c_1 \overset{3}{—} c_2 \overset{4}{———} c_3 \overset{3}{———} c_4$

A = [0 3 4 7 9 10]

K = 4   Ans = 3

K	Max possible Min dist.
2 →	10
3 →	4
4 →	3
5 →	1
6 →	1

Max possible min distance ∝ 1/K → BS on answer

```
// Define search space
l = 0                r = A[N-1] - A[0]
while (l <= r) {
    mid = l + (r-l)/2
    // Check mid
    cnt1 = cows (A, mid)
    cnt2 = cows (A, mid +1)
    if (cnt1 >= K  &&  cnt2 < K)   return mid
    // Decide left / right
    if (cnt1 < K)   r = mid-1
    else    l = mid+1
```

```
}
```

$TC = \underline{O(N * \log(A[N-1] - A[0]))}$ $SC = \underline{O(1)}$