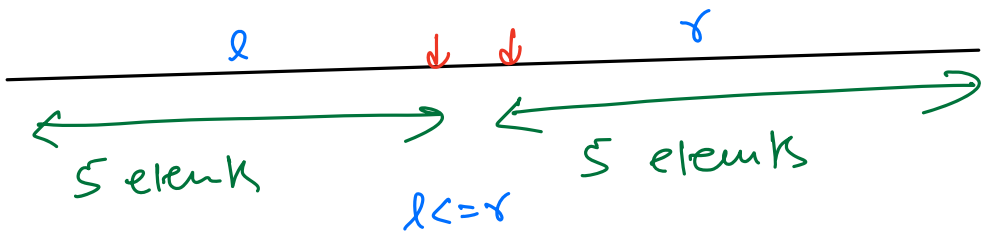
Searching 3 : Binary Search on Answer

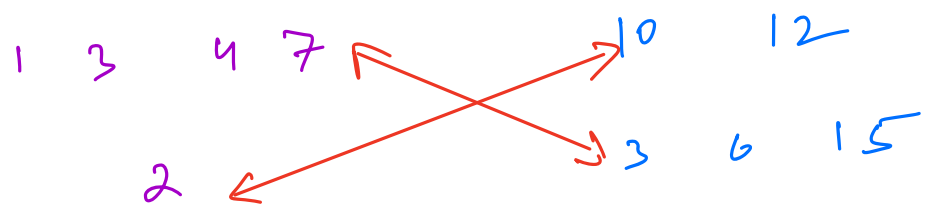
find median of 2 sorted arrays contd.....

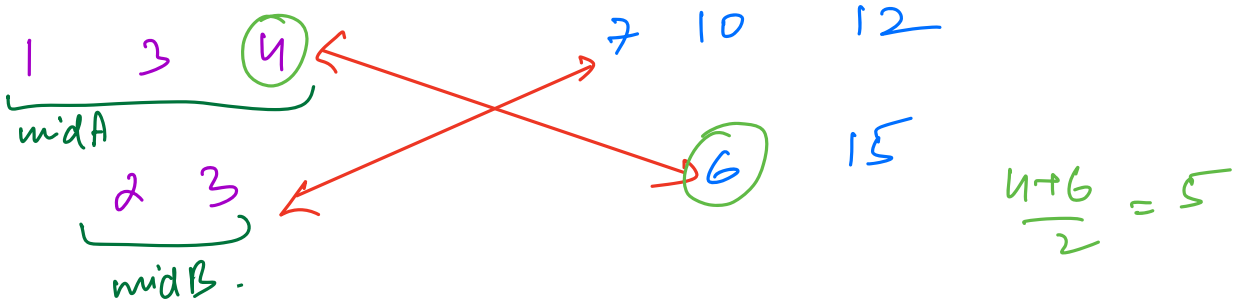
A = [1 3 4 7 10 12] # = 6

10 elements

B = [2 3 6 15] # = 4







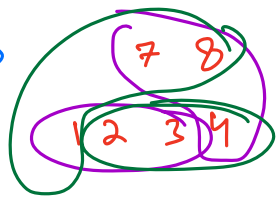
let's say array A is smaller than B.
↳ size N ↳ size M

define search space

l = 0, r = n

while (l <= r) {

// check if mid is answer



```
midA = (l+r)/2
midB = (n+m+1)/2 - midA

max_leftA  = (midA == 0) ? INT_MIN : A[midA - 1]
min_rightA = (midA == n) ? INT_MAX : A[midA]

max_leftB  = (midB == 0) ? INT_MIN : A[midB - 1]
min_rightB = (midB == m) ? INT_MAX : A[midB]

if ( max_leftA <= min_rightB &&
     max_leftB <= min_rightA ) {

    if( (n+m) % 2 == 0 ) {

        return  max( max_leftA, max_leftB) +
                min( min_rightA, min_rightB)
                ------------------------------
                            2.0

    }
    else {
        return max( max_leftA, max_leftB)
    }
}

// decide whether to go left or right

if( max_leftA > min_rightB )
    r = midA - 1
```

```
else
    l = mid A + 1)

}
```

TC = O(log N)

or

SC = O(1)

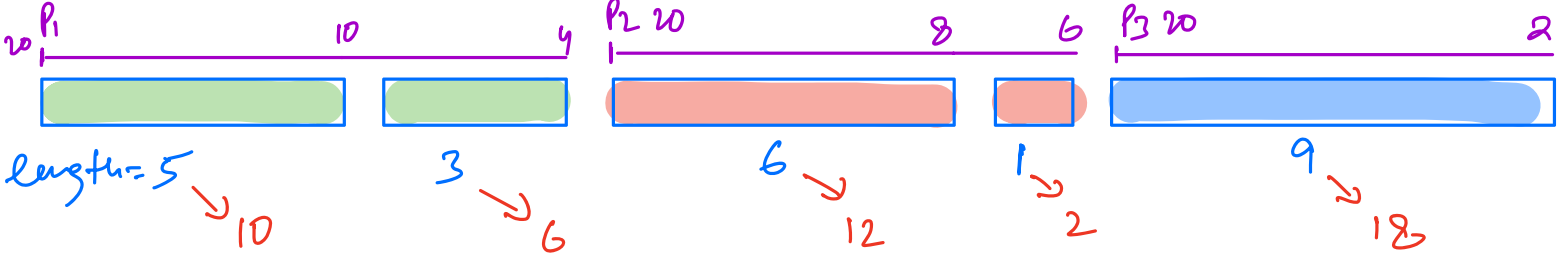
O(log (min(N,M)))

Question 1 → Painter Partition problem

Given N boards with length of each board.

1. All painters take T unit of time to paint 1 unit of length.
2. A board can only be painted by 1 painter.
3. A painter can only paint boards placed next to each other.

Find min. no. of Painters required to paint all boards in X unit of time. Return -1 if not possible.



T = 2

X = 15 → ans = -1

1 Painter = 10 + 6 + 12 + 2 + 18 = 48

>= 5 Painters = 18

X = 20 → ans = 3

X = 40 → ans = 2

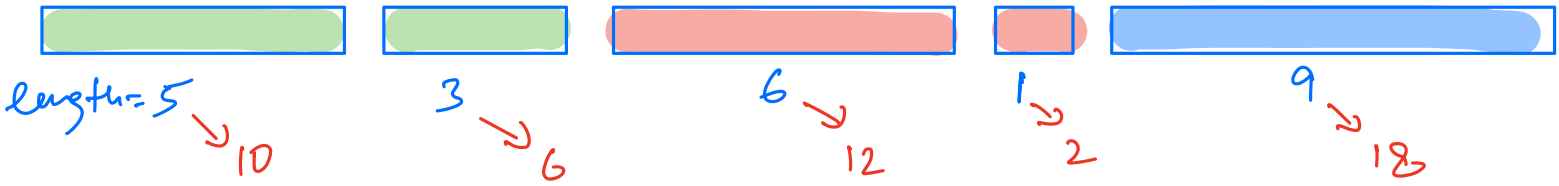
```
cnt=1 , p_time = X
for ( i=0 to n-1) {
    b_time = A[i] * T      // A[i] → length
    if (b_time > X)  return -1
    if ( p_time >= b_time)  p_time -= b_time
    else {
        cnt++
        p_time = X - b_time
    }
}
return cnt
```

TC = O(N)

SC = O(1)

Part 2

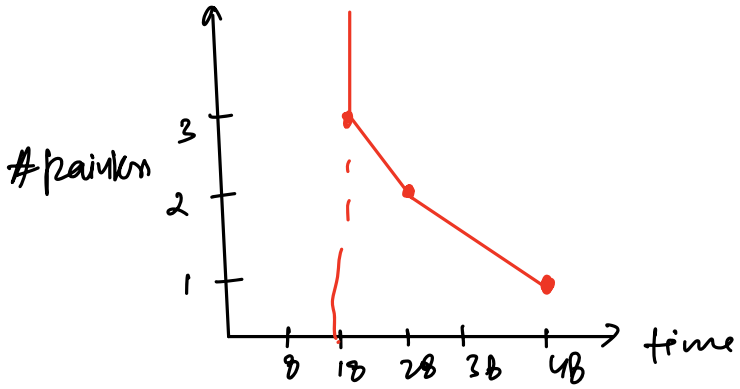
Find min. time to paint all boards if P painters are available. (P > 0).



Painters	Time
1	10+6+12+2+18 = 48
2	min{ max(10, 6+12+2+18) → 38
	max(10+6, 12+2+18) → 32
	max(10+6+12, 2+18) → 28
	max(10+6+12+2, 18)} → 30

>=3

18



time ∝ 1/p → binary search on answer

find min time?

// Define search space

$l = \max_{\forall i}(A[i]) \times T$ $\quad r = (\sum_{i} A[i]) \times t$

```
while( l <= r) {
// check if mid is answer
    mid = l + (r-l)/2        // mid -> time
    cut  = minPainters (mid, A)  -> TC=O(N)
    cut1 = minPainters (mid-1, A)
    if( cut == P && cut1 > P)
        return mid

// decide whether to go left or right
    if ( cut <= P)
        r = mid-1

    else
        l = mid+1

}
```

$TC = O(N * \log(\sum A[i] \times t))$

$SC = O(1)$

Question 2 → Aggressive cows

A farmer has N stalls.

A[i] → location of ith stall in ascending order

cows are aggressive towards each other. So, the farmer wants to maintain min D distance b/w any pair of cows.

find max # cows the farmer can have.

1. In a stall only 1 cow can be present
2. All cows have to be placed

	0	1	2	3	4	5	6
A = [	1	3	4	8	10	12	17]
	✓	x	x	x	x	✓	x
	✓	x	x	✓	x	x	✓

D = 10 ans = 2

D = 7 ans = 3

D = 4 ans = 4

```
cut=1 , L = A[0]

for ( i=1 to n-1) {
    if (A[i] - L >= D) {
        cut++
        L = A[i]
    }
}
return cut
```

TC = O(N)

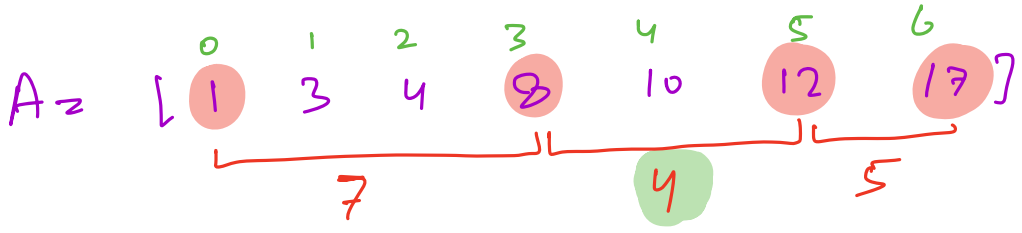
SC = O(1)

Part 2

Cows are aggressive towards each other so the farmer want to maximize the minimum distance b/w any pair of cows.

Find max possible min distance.

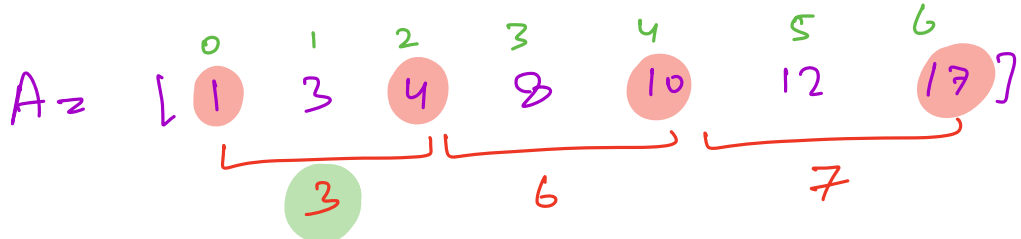
No. of cows = C (>= 2)



C = 4

ans = 4

4 > 3



cows dis b/w each pair

$C \propto 1/D$ → binary search on answer

Define search space

```
l = 1          r = A[n-1] - A[0]

while ( l <= r ) {
  // check if mid is the answer
     mid = l + (r-l)/2        // mid = distance

     cut = maxCows(mid, A)  → TC = O(N)
     cut1 = maxCows(mid+1, A)

     if ( cut == C && cut1 < C )
          return mid

  // decide whether to go left or right

     if ( cut >= C )    l = mid+1

     else               r = mid-1

}
```

TC = $O(N * \log(A[n-1] - A[0]))$

SC = $O(1)$