Today's Content

Painters Partition
Aggressive cows.

Today's Quote

A dream without clarity is
a nightmare.

Painters Partition Problem.

Ques

we have to paint n boards of length $A_1, A_2 \ldots A_n$. There are k painters available and each takes 1 unit of time to paint 1 unit of board.

find min time to get the job done.

→ 1 painter will paint only continuous sections of the board.

→ Two Painters can't share a board.

e.g 1) $A = \{\underbrace{10, 10}_{1}, \underbrace{10, 10}_{2}\}$ Ans = 20.

$k = 2$

e.g 2) $A = \{\underline{10, 20, 30}, \underline{40}\}$, Ans = 60.

$k = 2$

Ex

N = 15 .   k = 3 .

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

40 mins

25 mins

20 mins .

idea 1 :- divide the work into $\frac{\text{totaltime}}{k}$.

arr[6] = 1 1 1 2 1 100   k = 2

$\frac{\text{sum}}{k} \Rightarrow \frac{106}{2} = 53$.   Ans = 100 .

Not possible to divide like this .

Ex

N = 15 .   k = 4 ..

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

$P_1 = 26$ $P_2 = 30$ $P_3 = 15$

P_1 P_2 P_3 P_4

Can we do all task in 30 mins .

30 ✓ 31 ✓ 32 ✓

Can we finish all tasks in 10 mins?.

8 9 10

× × ×

8 9 10 . . x . . . 30 31 32

× × × ✓ ✓ ✓ ✓

Target :- min time to paint all the boards

SearchSpace :- lo → max (Array)

hi → sum (Array)

Ex

N=15 . k=4

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

$w_1=16$ $w_2=18$ $w_3=15$ $w_4=16$

$w_1=16$ $w_2=18$ $w_3=22$ $w_4=15$

$w_1=16$ $w_2=18$ $w_3=15$ $w_4=16$

$w_1=16$ $w_2=15$ $w_3=14$ $w_4=16$

$w_1=24$ $w_2=21$ $w_3=20$ $w_4=6$

$w_1=34$ $w_2=37$

lo	hi	mid		
10	71	40	Ans = 40	goto left
10	39	24	Ans = 24	goto left
10	23	16	goto	right
17	23	20	goto	right
21	23	22	Ans = 22	goto left
21	21	21	goto	right
22	21		Break	

Pseudo code :-

```
int  workers (int time[], int n, int k) {
    int lo = maxEle(arr);
    int hi = sum(arr);
    while (lo <= hi) {
        m = lo + (hi - lo)
                      2
        if (check (m, time, k)) {
            ans = m;
            hi = mid-1
        }
        else {
            lo = mid+1
        }
    }
    return ans;
}
```

T.C → O (log (sum - max) * n)

S.C → O(1)

0	1	2		4	5	6	7	8	9	10	11	12	13	14
3	5	1	7	8	2	5	3	10	1	4	7	5	4	6

```
bool check (int m, int time[], int n, int k){
        s=0;   c=1;
        for(i=0; i<n; i++){
            s= s+ time[i];
            if (s > m) {
                c++;
                s= time[i];
                if( c>k) {
                    return false;
                }
            }
        }
        return True;
}
```

Aggressive Cows

Ques) Given N cows & M stalls, all M stalls are on x axis at diff. locations, place all N cows such a way that min dist b/w any 2 cows is Maximized.

Note 1:- In a stall only one cow can be present.

Note 2:- All cows have to be placed.

Ex 1:-

	0	1	2	3	4	
	1	2	4	8	9	Min dis
	C_1	C_2	C_3			1
Stalls = 5	C_1		C_2		C_3	3
Cows = 3	C_1			C_2	C_3	1

Ex 2

Stalls = 9

cows = 4

0	1	2	3	4	5	6	7	8	
2	6	11	14	19	25	30	39	43	
C_1	C_2	C_3	C_4						3
C_1		C_2		C_3		C_4			8
C_1			C_2			C_3		C_4	12

Stalls = 9

cows = 4

0	1	2	3	4	5	6	7	8
2	6	11	14	19	25	30	39	43
C_1					C_2			
C_1		C_2		C_3	C_4			

check if we can place cows at min 20 distance away.

→ 20 21 22 23

20	21	22	23
✓	×	×	×

3	4	5	~~X~~	×	×	20	21	22	23
✓	✓	✓	–			×	×	×	×

Can we place cows atleast 5 dist. away?

3 4 5

Target → min dist b/w any two cows should be max.

Search space → lo = min dis b/w two adj. elements.

hi = max - min

cows = 4.

0	1	2	3	4	5	6	7	8
2	6	11	14	19	25	30	39	43

C_1 ———— C_2

C_1 ———— C_2 ———— C_3 ———— C_4

C_1 ———— C_2 ———— C_3

C_1 ———— C_2 ———— C_3

C_1 ———— C_2 ———— C_3

lo	hi	mid	
3	41	22	goto left
3	21	12	Ans = 12 goto right.
13	21	17	goto left
13	16	14	goto left
13	13	13	goto left.
13	12	break	

Ans → 12.

Pseudo Code :-

```
int  moo ( int dist[], int n, int c) {
        lo = min (dist[i+1] - dist[i])  ∀ i
        hi = arr[n-1] - arr[0];
        while (lo <= hi) {
              m = lo + (hi-lo)/2

              if (check (m, dist, c)) {
                    ans = m;
                    lo = mid+1;
              }
              else {
                    hi = mid-1
              }
        }
        return ans;
}
```

0	1	2	3	4	5	6	7	8
2	6	11	14	19	25	30	39	43

last_placed

```
bool check (int m, int dist[], int n,
                               int cows)

    last_placed = dist[0];
     count = 1;
     for (i=1; i<n; i++) {
         if (dist[i] - lastPlaced >= m) {
                   c++;
                   last_placed = dist[i];
                    if (c == cows) {
                          return True;
                    }
         }
     }
     return false;
}
```

Task to do

1- 2 constraint

(k painters)

(c cows)

1 minimize or maximize.

Search Space :- whatever we are asked to minimize or maximize.

Binary search on Answer.

T T T T T T T T f f f f f (Agg. cows)

f f f f f f f f T T T T T T T T (Painters Partition)

monotonic nature.

→ Problem is feasible till a point and not after that or vice versa.

$x^2 \leq n$ sqrt so

largest n, floor(sqrt) n

1	2	3	4	5	6	7	8	9	10	10
T	T	T	T	T	T	T	F	F	F	F

Category,

5th, 2, 3

1 2 3 4 5 6 7 8 9 10 x x x

F F F

x at which the no. of ~~~ we <= 5,

Book Allocation Problem

$$\begin{bmatrix} 7 & 3 \\ 2 & 1 \\ 4 & 9 \end{bmatrix}_{3\times 2}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow [1, 5, 9]$$

4 4

4

(1, 4, 9)

3 5

$$\begin{bmatrix} 1 & 2 & 3 \\ 5 & 9 & 13 \\ 4 & 9 & 20 \end{bmatrix}$$

1 10 4

win

[3, 5 ,]

1 1 2

2