Lecture : Binary search on answers.

Agenda

- Painter Partition problem.
- Aggressive cows.

Lecture starts at 7:05AM

Painter Partition problem.

Qu. We have to paint n boards of length $A_1\ A_2 \text{----} A_n$.

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.

constraint: 1 painter will paint only continuous section of boards.

Example: A = {10 10 10 10} K = 2

$\underbrace{10}_{P1}\ \underbrace{10\ \ 10\ \ 10}_{P2} \longrightarrow$ 30 units

$\underbrace{10\ \ 10}_{P1}\ \underbrace{10\ \ 10}_{P2} \longrightarrow$ $\boxed{\text{20 units}}$ Ans

$\underbrace{10\ \ 10\ \ 10}_{P1}\ \underbrace{10}_{P2} \longrightarrow$ 30 units

Ex2: A = [10 20 30 40] K = 2.

$\underbrace{10}_{P1}\ \underbrace{20\ \ 30\ \ 40}_{P2} \longrightarrow$ 90 units

$\underbrace{10\ \ 20}_{P1}\ \underbrace{30\ \ 40}_{P2} \longrightarrow$ 70 units

$\underbrace{10\ \ 20\ \ 30}_{P1}\ \underbrace{40}_{P2} \longrightarrow$ $\boxed{\text{60 units}}$ Ans

Ex3: $k = 3$

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

P1 = 34 | P2 = 37 — 40 mins ✓

P1 = 34 | P2 = 37 — 37 mins ✓

P1 = 24 | P2 = 25 | P3 = 22 — 25 mins ✓

P1 = 16 | P2 = 18 | P3 = 15 — 20 mins ✗

17	18	19	20 mins	....	37 mins	38	39	40	50
✗	✗	✗	✗		✓	✓	✓	✓	✓

Approach1: [Wrong Approach]

Example: $A = [\underbrace{1 \quad 2 \quad 3}_{P1} \quad \underbrace{4 \quad 5}_{P2} \quad \underbrace{6}_{P3} \quad 7]$

$k = 3$

Total work = $1 + 2 + 3 \ldots\ldots 7 = 28$

1 painter = $\frac{28}{3}$ = 9 units

Approach2:

A = [3 5 1 7 8 2 5 3 10 1 4 7 5 4 6]

k = 4.

Target = min time to get the board painted.

Search space: min and max possible time

s :- arr: [$\underbrace{1}_{p1}$ $\underbrace{2}_{p2}$ $\underbrace{3}_{p3}$ $\underbrace{4}_{p4.}$]   k = 4, ... ∞

max(arr)

e :- arr: [$\underbrace{1 \quad 2 \quad 3 \quad 4}_{p1 = 10}$]   k = 1

sum(arr)

Condition and dry run

A = [3 5 1 7 8 2 5 3 10 1 4 7 5 4 6]

K = 4.

P1 = 34 P2 = 36

P1 = 24 P2 = 21 P3 = 20 P4 = 6

P1 = 16 P2 = 15 P3 = 14 P4 = 16

P1 = 16 P2 = 18 P3 = 15 P4 = 20

P1 = 16 P2 = 18 P3 = 22 P4 = 15

s	e	mid		
10[max]	71(sum)	40 (yes)	ans = 40	e = mid − 1
10	39	24 (yes)	ans = 24,	e = mid − 1
10	23	16 (no)	s = mid + 1	
17	23	20 (no)	s = mid + 1.	
21	23	22 (yes)	ans = 22,	e = mid − 1
21	21	21 (no)	s = mid + 1	
22	21	→ break		

```
int workers(int[] arr , int k) {
        int n = arr.length;
O(n) —— int s = getMax(arr);
O(n) —— int e = getSum(arr);
        ans = 0;
        while( s <= e ) {
            mid = s + ((e-s)/2);
            if ( isPossible( mid , arr, k)) { —— O(n)
                ans = mid;
                e = mid-1;
            } else {
                s = mid +1;
            }
        }
    return ans;
}
```

$\log_2(sum - max)$

TC: $O(n) + O(n) + O(\log_2(sum-max)) * O(n)$

leading

TC: $O(n) * \log_2(sum-max)$;
SC: $O(1)$

```
boolean isPossible(int time, int[] arr, int k) {
    sum=0;  cnt=1;
    for(int el: arr) {
        sum += el;
        if (sum > time) {
            cnt+=1;
            sum= el;
            if (cnt > k){
                return false;
            }
        }
    }
    return true;
}
```

Aggressive cows

Qu Given n cows and m stalls. All m stalls are on x axis at different locations, place all n cows such a way that min distance b/w 2 cows is maximised.

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

Ex1 arr: [1 2 4 8 9]

stalls = 5
cows = 3

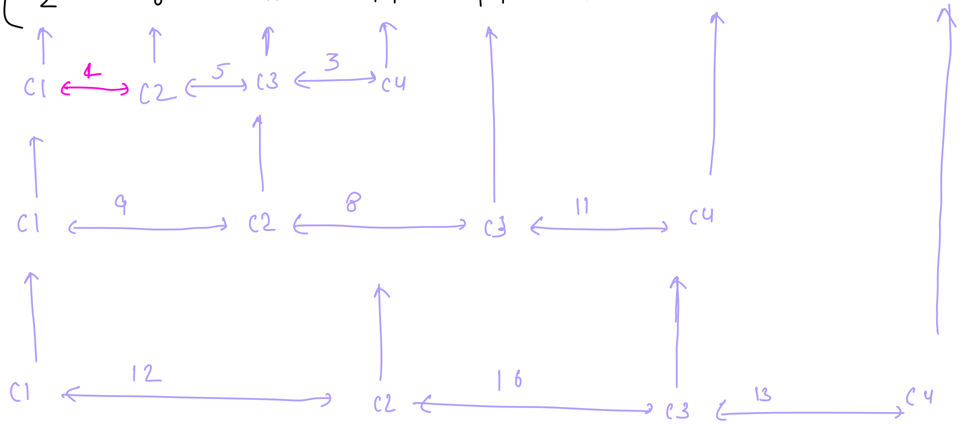
$C1 \xleftrightarrow{1} C2 \xleftrightarrow{2} C3$ 1.

$C1 \xleftrightarrow{3} C2 \xleftrightarrow{5} C3$ 3 → Ans

$C1 \xleftrightarrow{7} C2 \xleftrightarrow{1} C3$ 1

Ex2: stalls = 9
cows = 4

arr: [2 6 11 14 19 25 30 39 43]



3 distance

8 distance

12 distance

Target = maximise dist b|w any 2 cows

Search space:

s ⟶ min distance b|w any 2 cows

arr: [1 3 8 12] ⟹ min b|w any 2 adjacent els

C1 C2 C3 C4

2

cows = 4

e ⟶ max distance b|w any 2 cows

arr: [1 3 8 12] ⟶ max(arr) − min(arr)

cows = 2

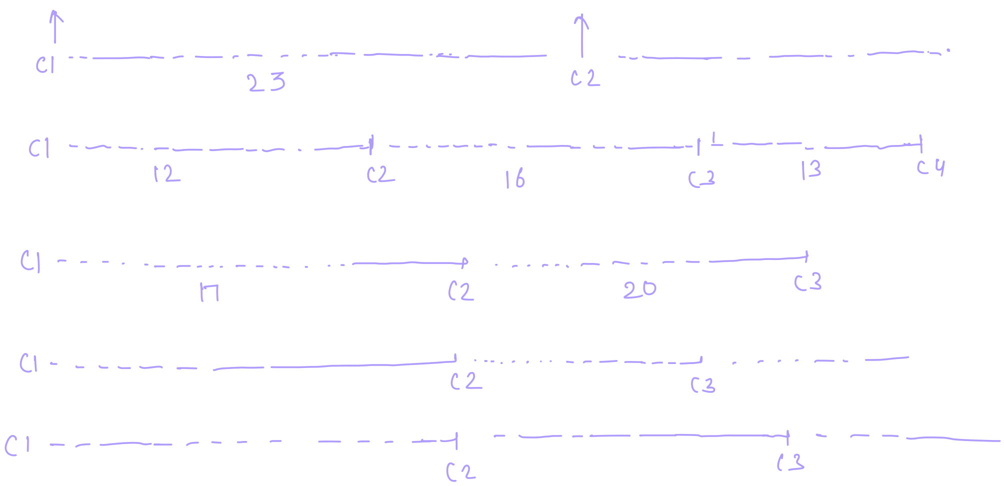
C1 ⟷ C2

11

Condition and dry run.

cows = 4

[2 6 11 14 19 25 30 39 43]



s	e	mid	
3	41	22	[Any two cows Dis >= 22] not possible e = mid − 1
3	21	12	ans = 12, s = mid + 1
13	21	17	not possible, e = mid − 1
13	16	14	not possible, e = mid − 1
13	13	13	e = mid − 1
13	12	break	

Code:

```
int moo( int[] dist, int n, int cows) {
         (sorted)
O(n) —— s = min dis b|w any 2 adjacent els
O(n) —— e = max(arr) - min(arr); // dist[n-1] - dis[0];

    ans = 0;
    while( s<=e){                                  O log2 (e-s)
        mid = s + (e-s)/2;
        if( isPossible( mid, dist, cows)) {  <—— O(n)
            ans = mid;
            s = mid+1;
        } else {
            e = mid-1;                 TC: O(n * log2(s-e))
        }                              SC: O(1)
    }
    return ans;
}

boolean isPossible( int mid, int[] dist, int cows){
    int last_placed_cow = dist[0];
    int cnt =1;
    for( i=1; i< dist.length; i++) {
        if( dist[i] — last_placed_cow >= mid) {
            cnt +=1;
            last_placed_cow = dist[i];
            if ( cnt == cows) {
                return true;
            }
        }
    }
    return false;
}
```

Thankyou ☺

Doubts:

cows = 2

mid = 12

12

9

4

2 6 11 14 19 25 30 39 43

c1 c2 c2 c2

last-placed-cow

last-placed-co

c2

cnt=1

n