Searching 3: Binary Search on Answer

TODAYS CONTENT

- Painters partition
- Aggressive cows

Question:- We have to paint all boards of length [C0, C1, C2, ... CN-1]. There are K painters available & each of them takes 1 unit of time to paint 1 unit of the board.

Calculate & return the minimum time required to get the job done.

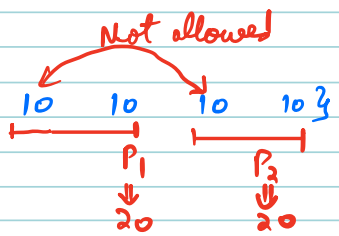
NOTE:-

(1) Two painter cannot share a board to paint. That is to say, a board cannot be painted partially by one painter & partially by another.

(2) A painter will only paint contiguous boards. This means a painter paints a continuous subarray of Boards.

Ex:- A = {10, 10, 10, 10}

K = 2



Ans = 20

Ex:- A = { 10, 20, 30, 40 }

K = 2

Ans ⇒ 60

IDEA :- Divide into $\frac{\text{Total time}}{K}$ i.e. each painter take $\frac{\text{Total time}}{K}$ unit of time.

Ex:- arr[] = { 1 2 3 4 100 } | k=2

$\frac{\text{Total time}}{K} \Rightarrow \frac{\cancel{104}\ 55}{2}$

→ Not possible

Quiz! :- what is the minimum time to get the job done?

A = [1 2 3 4 100] | k=2

↳ Ans = 100

↳ Binary Search

Ex

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

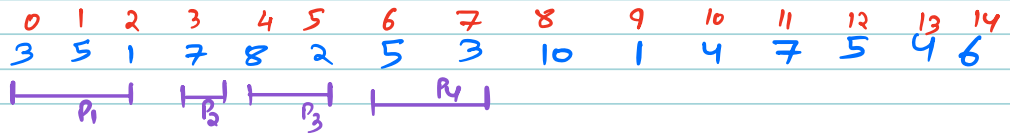
P1 P2 P3

Q Can all tasks completed in 30 Mins?

Lets Try, yes

Conclusion :- If jobs can be completed in 30 Mins then it can be completed in 31, 32, 33, ... also because each one will get more time

Ex



Q Can we finish all tasks in 10 Mins?

↳ No

Can we say it will be Not Completed in any time less than equal to 10

1 ..... 7 8 9 10 (x x x x)

IDEA?:- 1 ..... 7 8 9 10 ------- 30 31 32 33 .. (x x x x ✓ ✓ ✓ ✓)

⇒ Can we say definitely I will get Ans if I will keep increasing value from 10 which will smallest time for which we will get job done.

Now,

① Target :- Min time to get the job done.

② Search Space :- L = Max of Array.

Ex:- {1 2 3 4 100}

K=5 | Ans = 100

R = Sum of Array

Ex:- {1 2 3 4 100}

K=1 | Ans = 110

③ Condition to Descard

Case I :- when the work can be completed in 'x' time.

ans = X ← potential Ans

Goto Left

Case II :- when the work can't be completed in 'x' time.

Goto Right.

DRY RUN

Ex

L	R	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 23 22 break

PSEUDO CODE

```
int worker ( int[] time , N , K ){
    L = max (time)
    R = sum of time
    ans = R;
    while ( L <= R){
        int M = L + (R-L)/2
        if ( check (time , K, M)){
            ans = M
            R = M-1;
        } else {
            L = M+1;
        }
    }
    return ans;
}
bool check ( int[] time, K, mid){
    s=0 , c=1;
    for (i = 0 ; i < time.length; i++){
        s += time[i];
        if ( s > mid) {
            c = c+1;
            s = time[i];
            if ( c > K){
                return False;
```

TC - O(N x log(R-L))

SC - O(1)

```
        }
    }
    return true;
}
```

⇒ Binary Search problem identification

① These following type of problems generally has characteristics :-

→ There are two or 3 parameters & constraints

- 1-2 Constraints
- Maximize or minimize arrays

Eg :- ① Painters Partition
② Aggressive Cows
③ Allocate books.

② one Tricky point is to find search space which is generally the parameter asked to maximize or minimize.

⇒ Search Space :- See what is asked to maximize or minimize.

③ It follows monotonous behaviours

↳ T T T T T T F F F F T

↳ F F F F F T T T T T T T

Break : till 10:42

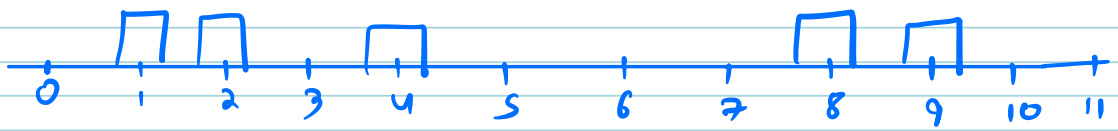
Question :- Given N cows & M stalls, all M stalls are located at the different locations at x-axis, place all the cows such that minimum distance b/w any two cows is maximized.

Note :- ① There can be only one cow in a stall at a time.

② We need to place all cows.

Eg

0	1	2	3	4	
1	2	4	8	9	C=3



		Min Dis
→	C_1 C_2 C_3	1
→	C_1 C_2 C_3	3
→	C_1 C_2 C_3	1

Ans = 3

★ Stalls should be in sorted order

Quiz 2 :- what is objective of the problem described?

↳ Place cows in stalls such that the minimum distance b/w any two cows is maximized.

Quiz 3 :- what will be the max value of the distance b/w the closest cows in this case?

A = [0, 3, 4, 7, 9, 10] | K = 4

↳ 3 Ans

Ex2:-

Stalls = 9

Cows = 4

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

Q Can I place atleast 8 distance away?

↳ yes, $c_1 = 2$, $c_2 = 11$

$c_3 = 19$, $c_4 = 30$

Q Can I place atleast 12 distance away?

↳ yes, $c_1 = 2$, $c_2 = 14$, $c_3 = 30$, $c_4 = 43$

Q Can I place atleast 20 distance away?

↳ No

So the ans can't be 21, 22, 23, ...

IDEA

..... 7 8 9 10 ------ 20 21 22

(✓ ✓ ✓ ✓ ... ✗ ✗ ✗)

So, the ans will be in the range.. And for that we need to decrease from 20

Binary Search

① **Search Space** → when cows == No. of stalls.

- what will be least possible distance?
  - ↳ L = Min Diff b/w two adjacent stalls
- what will be the max possible distance b/w stalls?
  - ↳ R = stalls[N-1] - stalls[0]
  - [Distance b/w first & Last]

② **Target** :- Maximum possible distance

③ **Condition to discard** [Already Discussed]

Ex :- [Cows = 4]

0	1	2	3	4	5	6	7	8
2	6	11	14	19	25	36	39	43

DRY RUN

L	R	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	

PSEUDO CODE

```
int Aggressive Cow (int[] stalls, N, Cows){
    L = min Dist. b/w two stalls.
    R = stalls[N-1] - stalls[0]
    ans = L;
    while (L <= R){
        int M = L + (R-L)/2;
        if ( check(stalls, N, Mid, Cows) ){
            ans = M;
            L = M+1;
        } else {
            R = M-1;
        }
    }
    return ans;
}
```

```
bool check ( stalls , N, mid, cows){
    last_placed = stalls [0];
    count = 1;
    for (i = 1; i<N; i++){
        if ( stalls [i] - last_placed >= mid){
            c= c+1;
            last_placed = stalls[i];
            if (c == cows){
                return True;
            }
        }
    }
    return False;
}
```

TC → O(N * log(R-L))

SC → O(1)