HW 07: Aggressive Cows (GFG & Code studio)

Problem Statement:
You are given an array consisting of **n** integers which denote the position of a stall.
You are also given an integer k which denotes the number of aggressive cows.
You are given the task of assigning stalls to k cows such that the minimum distance
between any two of them is the maximum possible.

Observation:
stall ki position = stalls[ith]
stalls array ka size = n
number of cows = k
mujhe find kya karna hai = minimum distance between two of them is maximum distance possible

---> **iska mtlb yeh hai ki** -- Pahle hame har ek case me two cows ke beech ka minimum distance find karna hai
uske baad hame minimum distances me se maximum distance as a final ans batana hai.

Example 01:
Input: n=5 , k=3, stalls = [1 2 4 8 9]
Output: 3

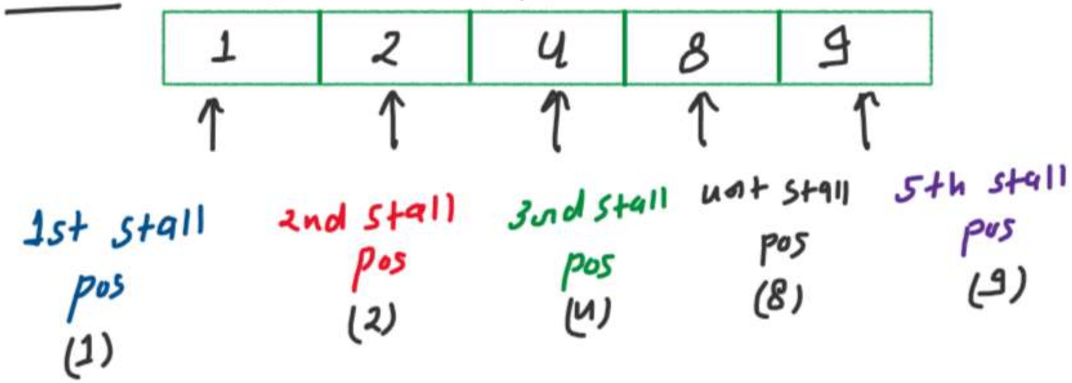
```
-------------------------------------------------------------------------------------------------------
Explanation:
1st stall ki position = stalls[0] = 1
2nd stall ki position = stalls[1] = 2
3rd stall ki position = stalls[2] = 3
4rt stall ki position = stalls[3] = 4
5th stall ki position = stalls[4] = 5

The first cow can be placed at stalls[0],
the second cow can be placed at stalls[2] and
the third cow can be placed at stalls[3].
The minimum distance between cows, in this case, is 3, which also is the largest among all possible ways.

-------------------------------------------------------------------------------------------------------
OPTIMAL APPROACH: Define search space and predicate function
Step 01: Sort the array
Step 02: Find Highest Minmum Distance to create search space's end point
Step 03: Now, Applying Binary Search on search space BinarySearch()
Step 04: create predicate function isPossibbleSol()

-------------------------------------------------------------------------------------------------------
Time Complexity: O(N*log(end)), Here N is size of array stalls and end is the highestMaxDis
Space Complexity: O(1), no extra space used

-------------------------------------------------------------------------------------------------------
Resource: https://practice.geeksforgeeks.org/problems/aggressive-cows/0
```

DRY RUN

Example 01:
Input: n=5 , k=3, stalls = [1 2 4 8 9]
Output: 3

STEP:01 SORTED ARRAY

1	2	4	8	9



STEP:02 Highest max Distance

End = 5th stall pos — 1st stall pos

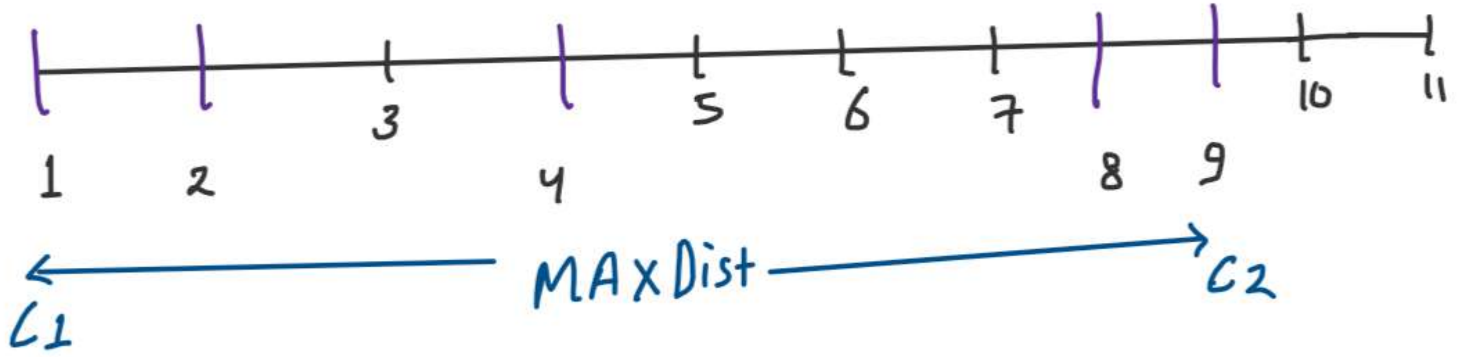
= stalls[4] — stalls[0]

= 9 — 1

= 8

8 ek aisa distance hai jis distance par two cow kabhi bhi nhi lad skti hai

positive Real No. Line



STEP:03 SEARCH SPACE

Iteration:01

start = 0

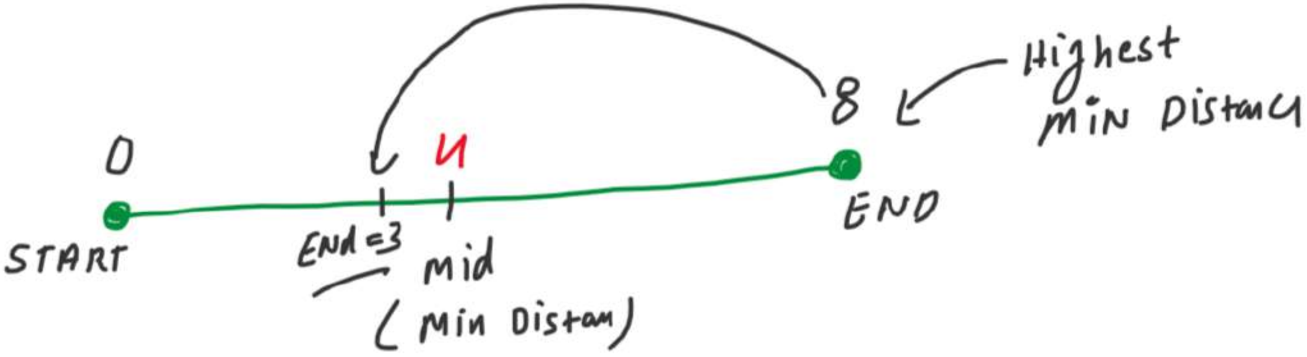
End = 8

mid = 0+8/2

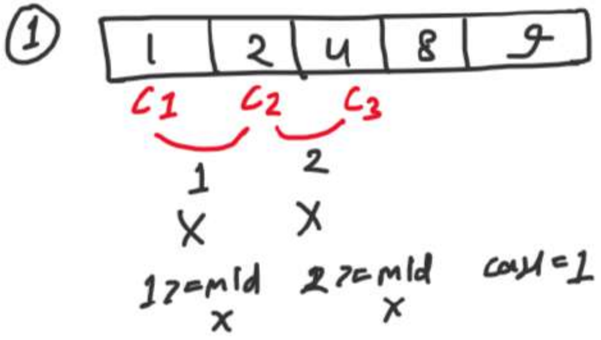
= 4

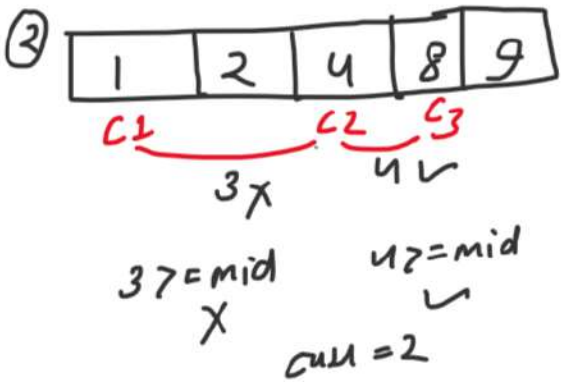
case = 1 | Ans = -1

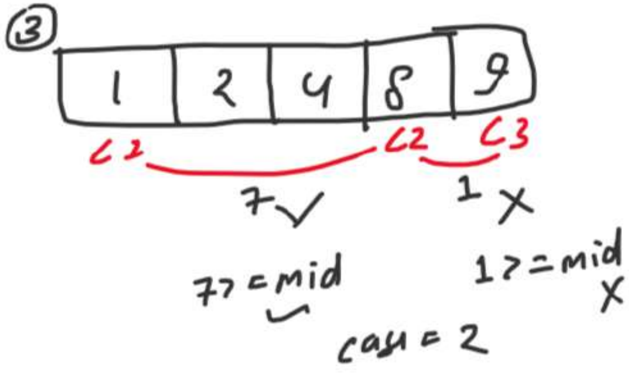
K = 3



STEP:4







Case==3 → NO Solo

X → end = mid-1

STEP:3

Iteration:02

start = 0

End = 3

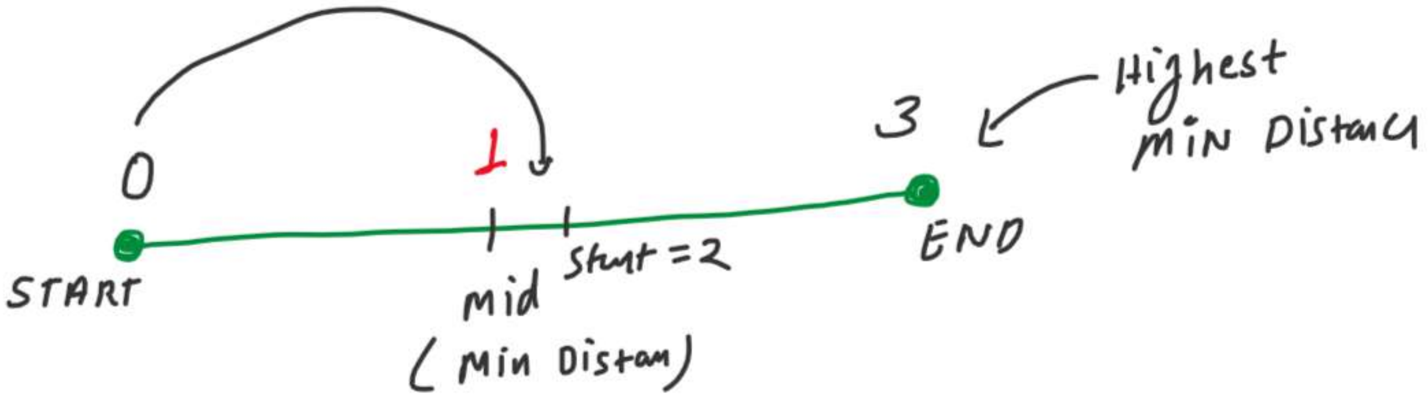
mid = $\frac{0+3}{2}$

= 1

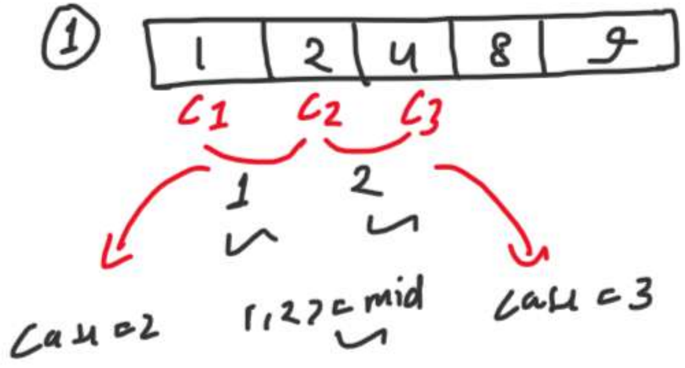
case = 1

K = 3

| Ans = ~~-1~~ 1



STEP:4



pos sol.n

Start = mid + 1

Ans = mid

STEP:3

Itaration:0 3

Start = 2

End = 3

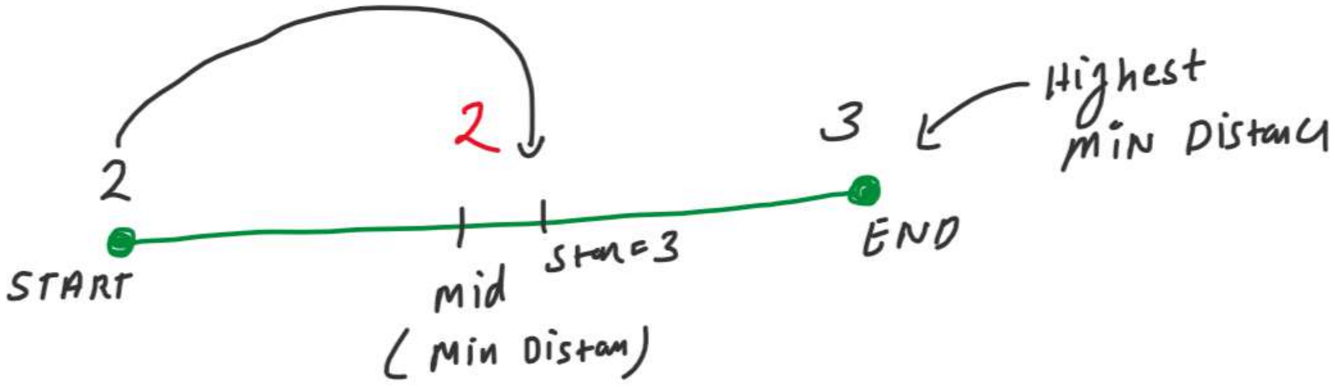
mid = $\frac{2+3}{2}$

= 2

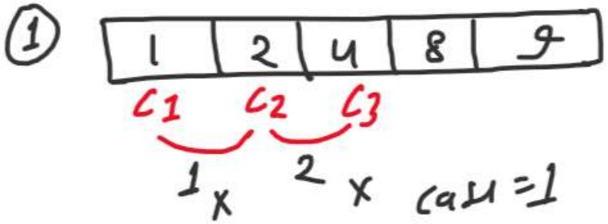
case = 1

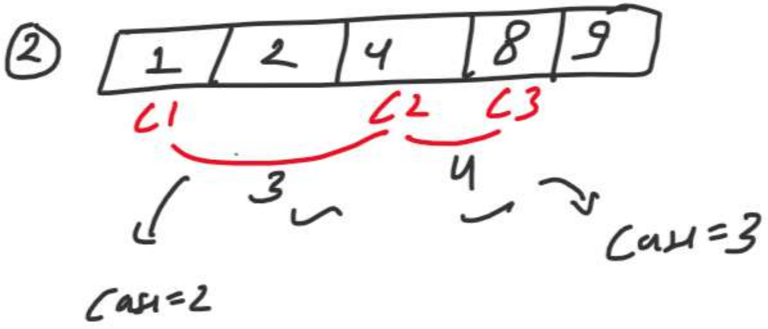
K = 3

| Ans = ~~1~~ 2



STEP:4





Pass sol.n

Ans = mid

start = mid+1

STEP:3

Itaration:04

Start = 3

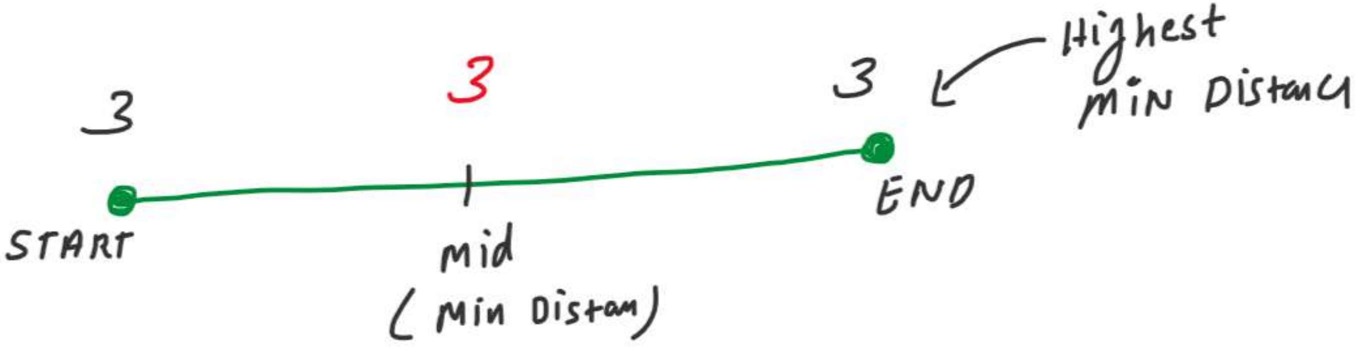
End = 3

mid = $\frac{3+3}{2}$

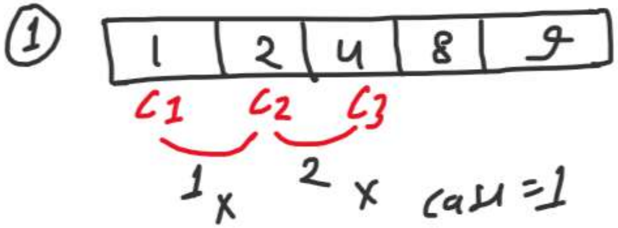
= 3

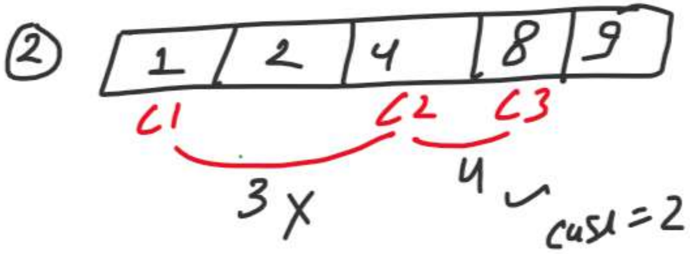
case = 1 | Ans = ~~2~~ 3

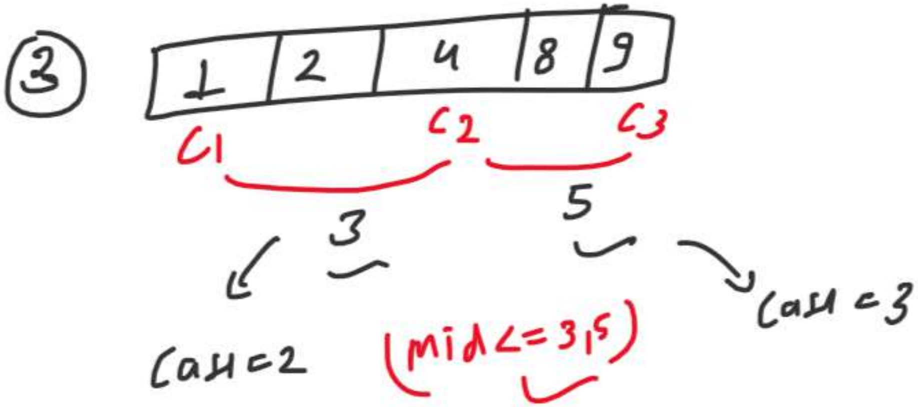
K = 3



STEP:4







Cows == 3

Return true → Ans = mid

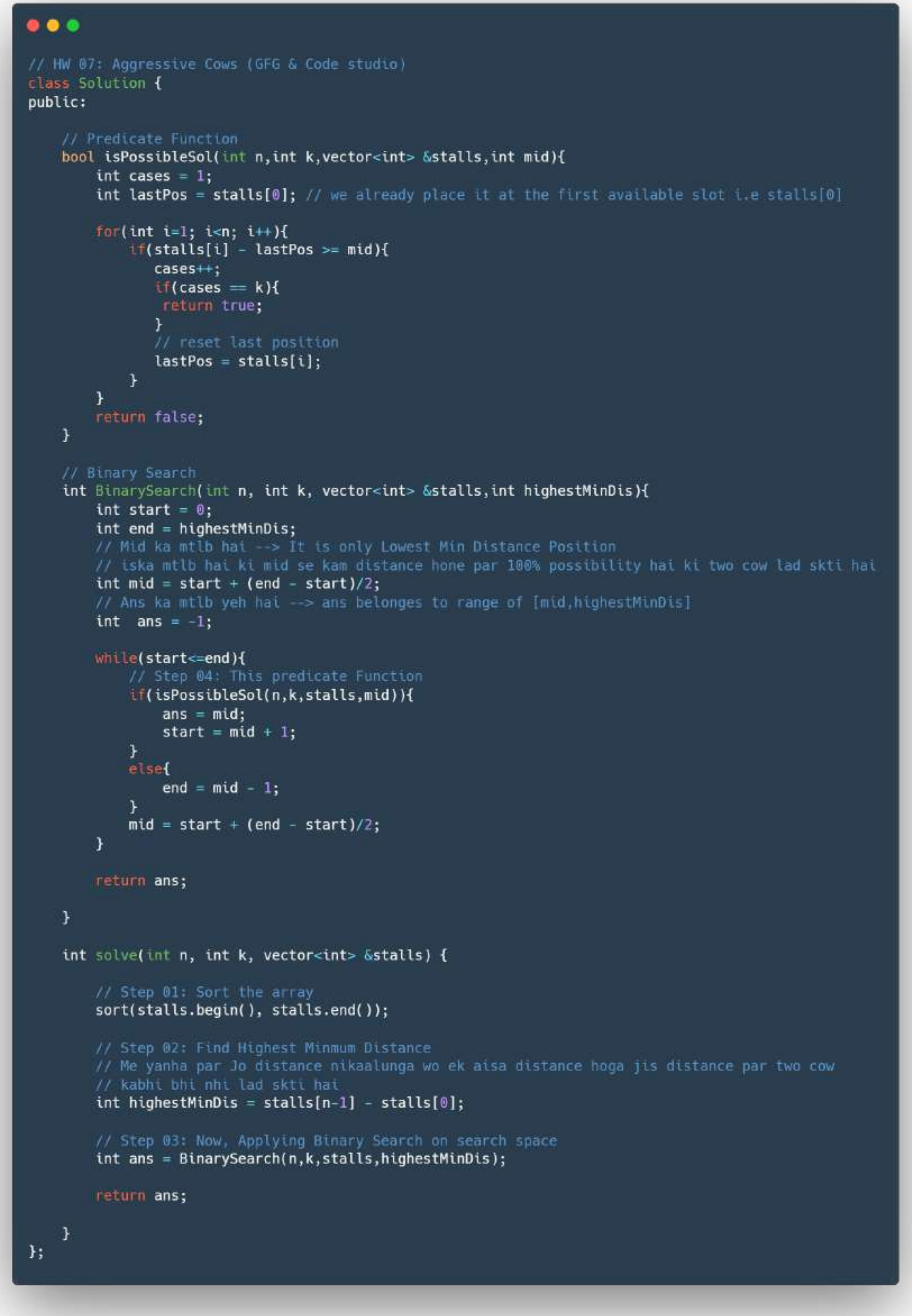
start = mid+1

Iteration : 05

$$\left.\begin{array}{l} start = 4 \\ End = 3 \end{array}\right\} \Rightarrow$$ RUK jaoo.... start > End

Final Ans = 3

Ans ka mtlb yeh hai --> ans belonges to range of [mid,highestMinDis]



```cpp
// HW 07: Aggressive Cows (GFG & Code studio)
class Solution {
public:

    // Predicate Function
    bool isPossibleSol(int n,int k,vector<int> &stalls,int mid){
        int cases = 1;
        int lastPos = stalls[0]; // we already place it at the first available slot i.e stalls[0]

        for(int i=1; i<n; i++){
            if(stalls[i] - lastPos >= mid){
               cases++;
               if(cases == k){
                return true;
               }
               // reset last position
               lastPos = stalls[i];
            }
        }
        return false;
    }

    // Binary Search
    int BinarySearch(int n, int k, vector<int> &stalls,int highestMinDis){
        int start = 0;
        int end = highestMinDis;
        // Mid ka mtlb hai --> It is only Lowest Min Distance Position
        // iska mtlb hai ki mid se kam distance hone par 100% possibility hai ki two cow lad skti hai
        int mid = start + (end - start)/2;
        // Ans ka mtlb yeh hai --> ans belonges to range of [mid,highestMinDis]
        int  ans = -1;

        while(start<=end){
            // Step 04: This predicate Function
            if(isPossibleSol(n,k,stalls,mid)){
                ans = mid;
                start = mid + 1;
            }
            else{
                end = mid - 1;
            }
            mid = start + (end - start)/2;
        }

        return ans;

    }

    int solve(int n, int k, vector<int> &stalls) {

        // Step 01: Sort the array
        sort(stalls.begin(), stalls.end());

        // Step 02: Find Highest Minmum Distance
        // Me yanha par Jo distance nikaalunga wo ek aisa distance hoga jis distance par two cow
        // kabhi bhi nhi lad skti hai
        int highestMinDis = stalls[n-1] - stalls[0];

        // Step 03: Now, Applying Binary Search on search space
        int ans = BinarySearch(n,k,stalls,highestMinDis);

        return ans;

    }
};
```

Example:02

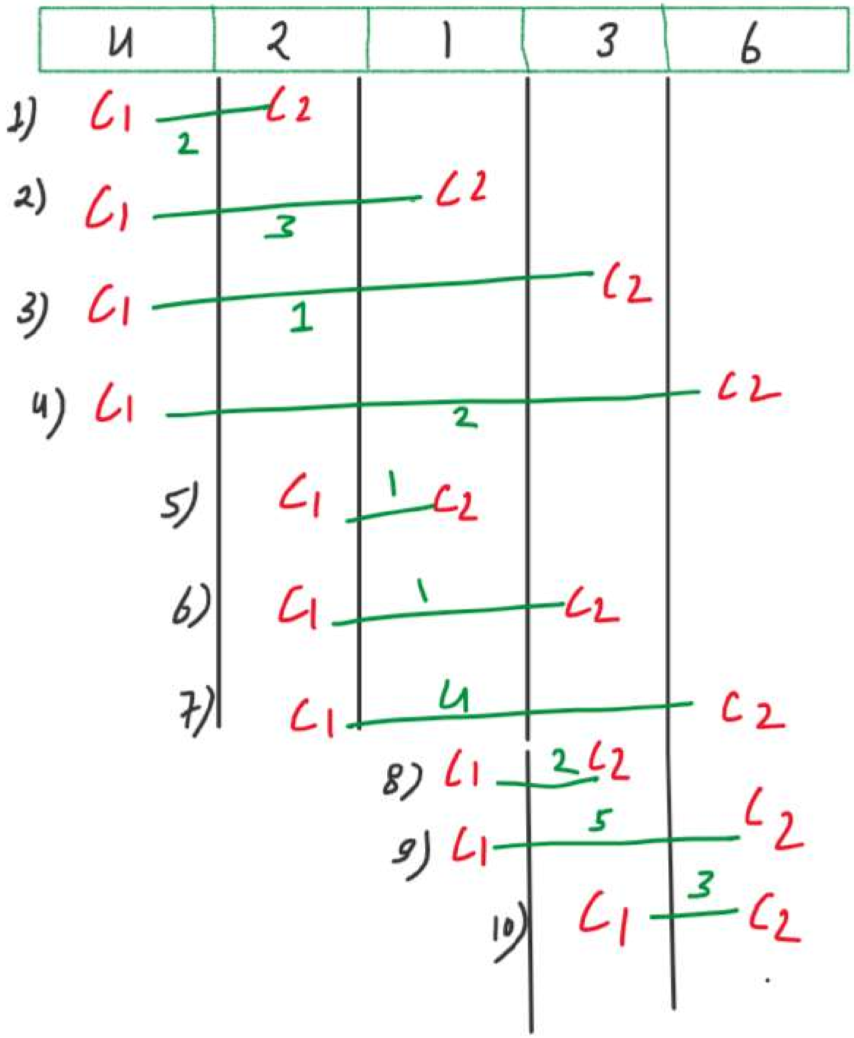
stalls = [4 , 2 , 1 , 3 , 6]

K (cows) = 2

N = 5

Output = 5

Explanation



Min Distance

2
3
1
2
1
1
4
2
5
3

Max Distance

5

Output

Total Combinations

$$\Rightarrow \frac{5!}{(5-2)! \times 2!}$$

$$\Rightarrow \frac{5 \times \overset{2}{\cancel{4}} \times \cancel{3 \times 2 \times 1}}{\cancel{3 \times 2 \times 1 \times 2 \times 1}}$$

$\Rightarrow$ 10 Combinations

DRY RUN

STEP: 01 $\boxed{\text{SORT}}$

stalls	1	2	3	4	6
	0	1	2	3	4

STEP: 02 $\boxed{\text{END}} = \text{stalls}[4] - \text{stalls}[0]$

$= 6 - 1$

$= 5$ → Highest min Distancy

STEP:03 SEARCH SPACE Binary Search

It:01

K=2

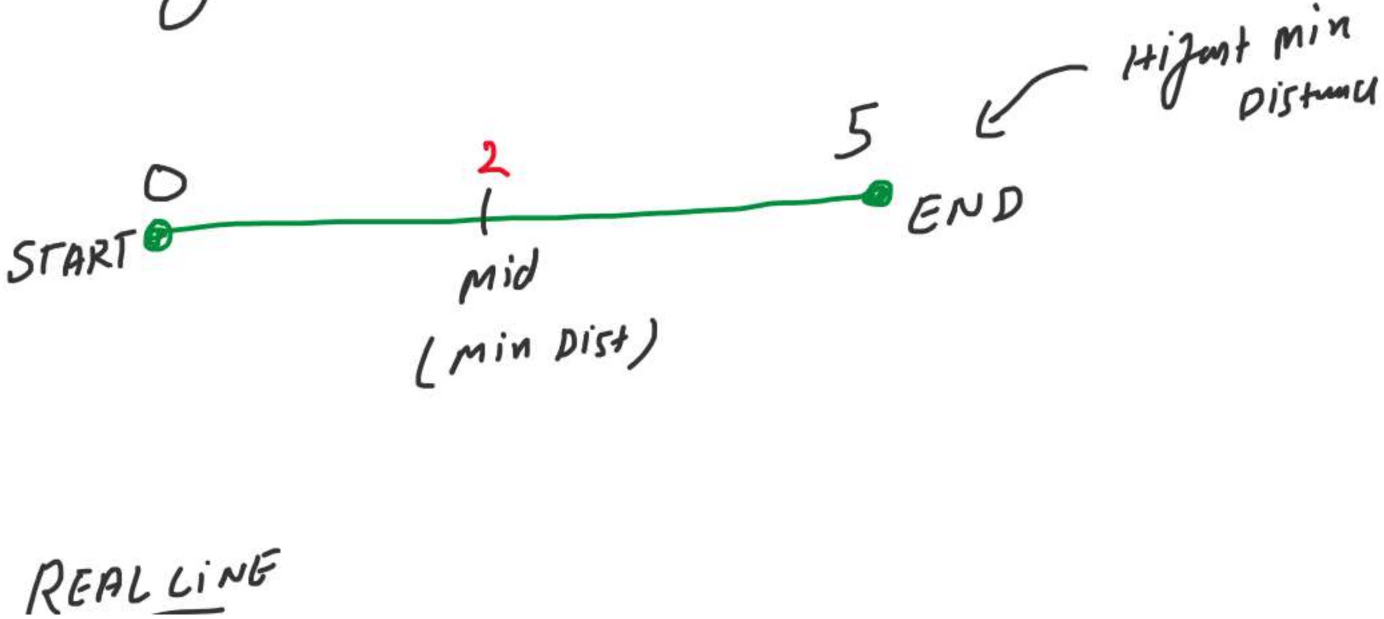
start = ~~0~~ 3

End = 5

mid = $\frac{0+5}{2} = 2$

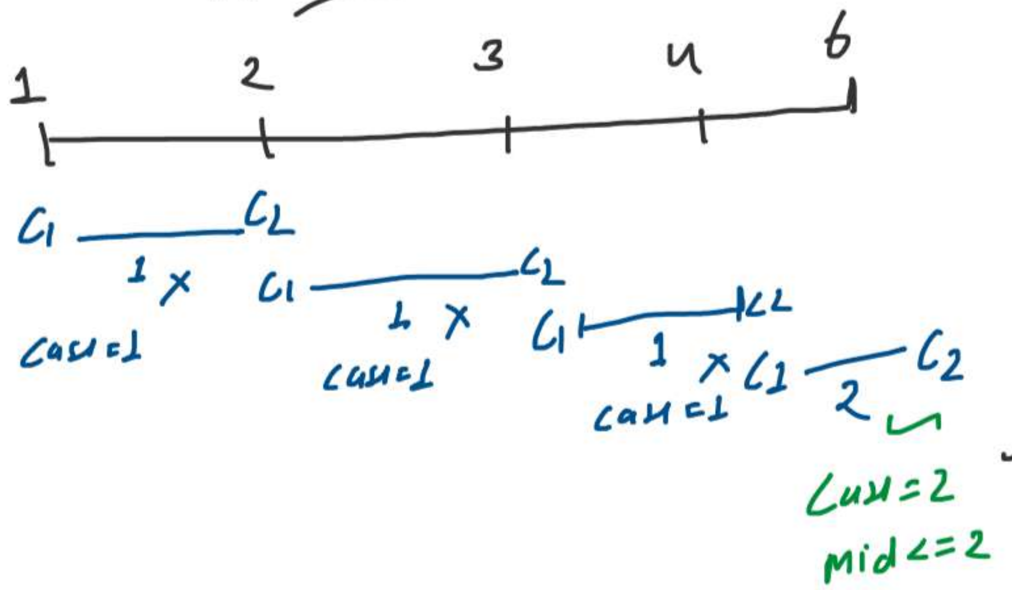
case = 1

ANS = ~~-1~~ 2



REAL LINE

STEP:04



Poss. sol.n

case == K
return true

Ans = mid
start = mid + 1

STEP:03 (SEARCH SPACE) Binary Search

It:02

K=2

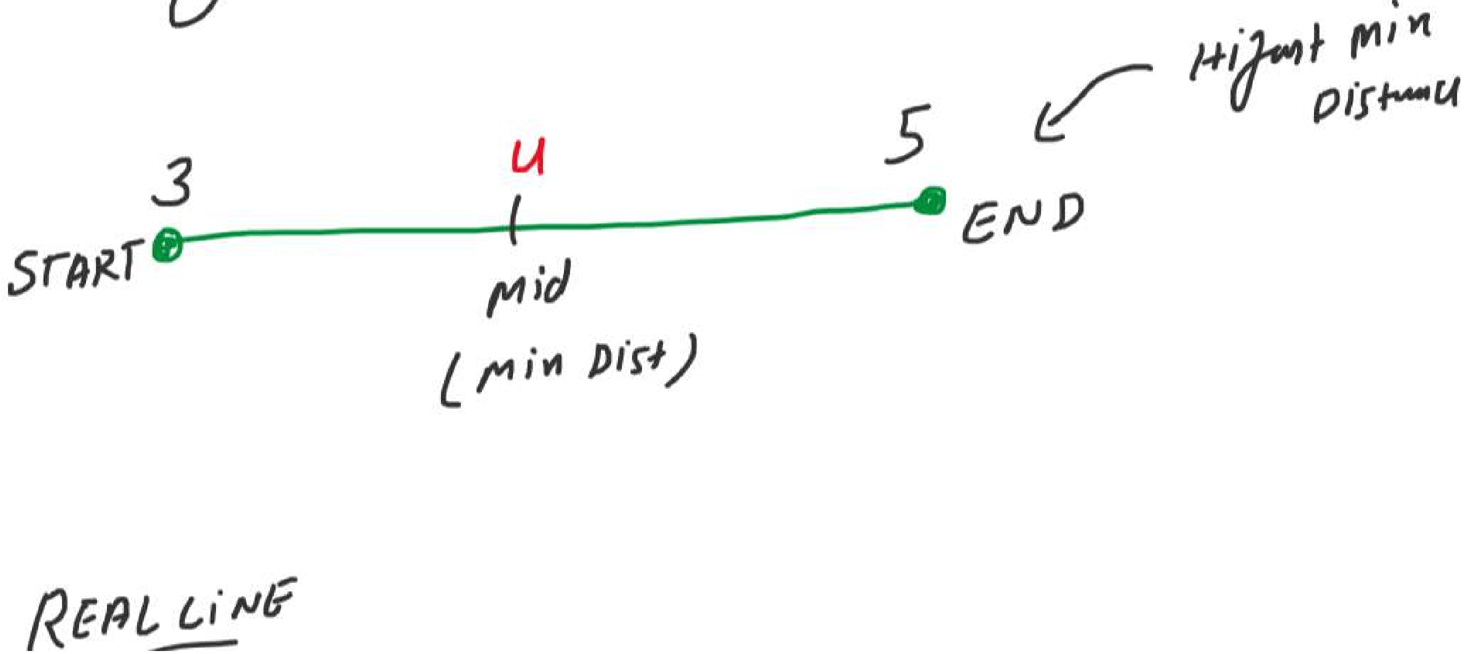
start = ~~3~~ 5

End = 5

mid = $\frac{3+5}{2} = 4$

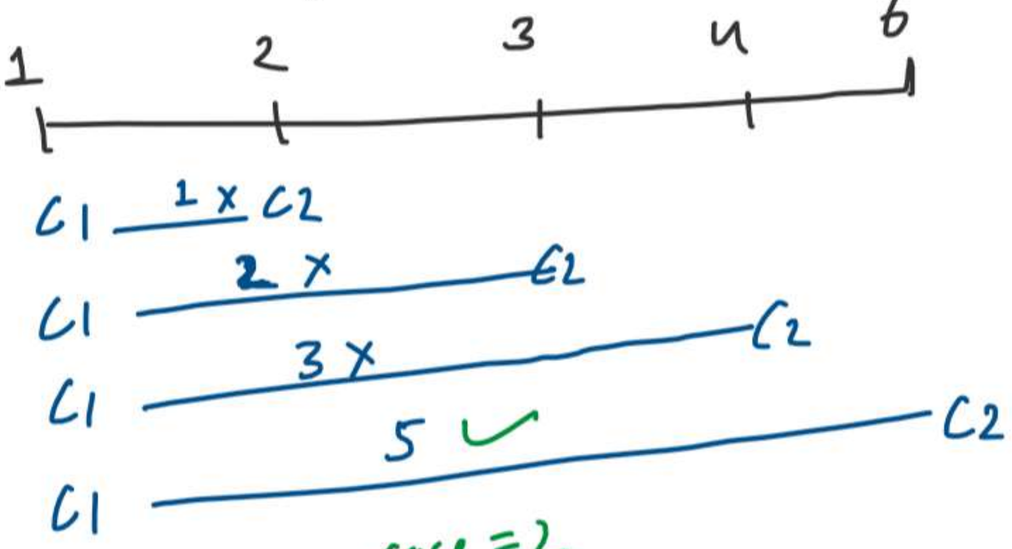
case = 1

ANS = ~~2~~ 4



REAL LINE

STEP:04



case = 2
mid <= 5

Poss. sol.^{n}

case == K
return true

Ans = mid
start = mid + 1

STEP:03 (SEARCH SPACE) Binary Search

It:03

K=2

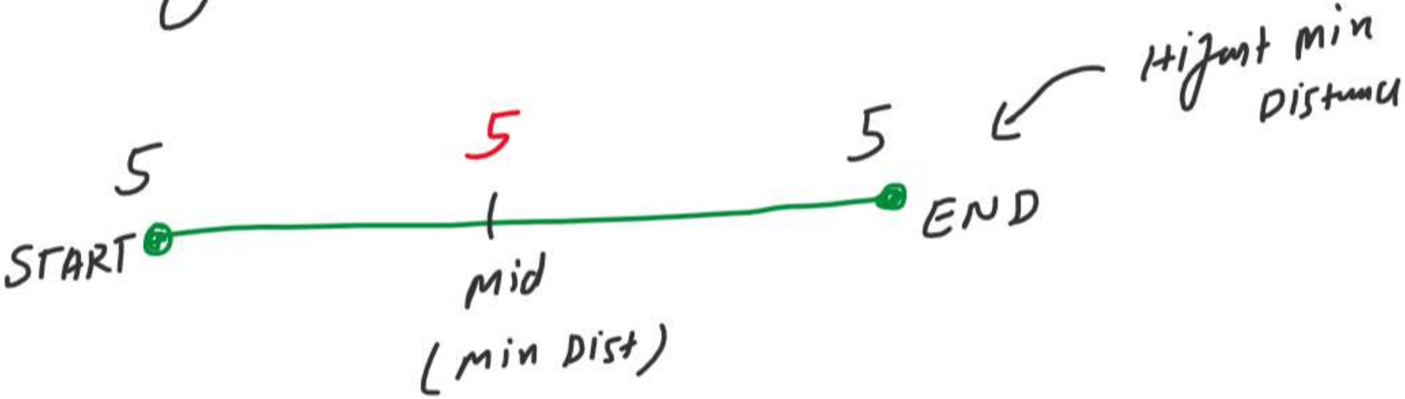
start = 5

End = 5

mid = 5

Case = 1

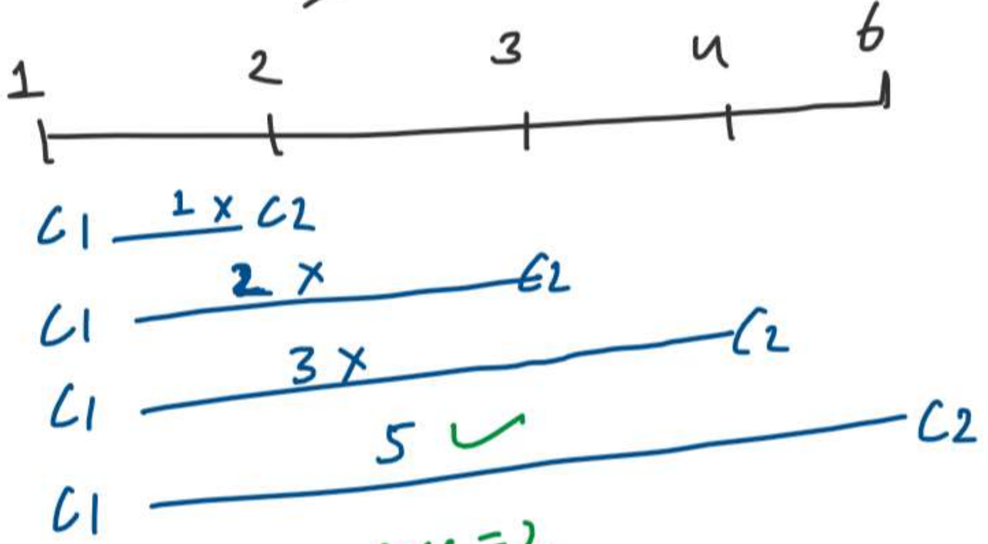
ANS = ~~4~~ 5



REAL LINE

STEP:04

REAL LINE



case = 2
mid <= 5

Poss. sol.n

case == K
return true

Ans = mid
start = mid + 1

Iteration: 4

$$\left.\begin{array}{l}\text{Start} = 6\\ \text{End} = 5\end{array}\right\}\ \underline{\text{STOP}}\qquad \text{start} > \text{End}$$

Output ⇒ 5