07/02/2026

Saturday

Aggressive cows (min dist between 2cows 7cows) is max

arr[] = [0 3 4 7 10 9] cows=4

first sort these there are all stalls (shops)

0 3 4 7 9 10

first $C_1 \xleftrightarrow{3} C_2 \xleftrightarrow{1} C_3 \xleftrightarrow{3} C_4$ min 1

Here all cows but we want max(min)

* U needs to place all cows in stalls in a way to get max(min)

0 3 4 7 9 10

$C_1 \xleftrightarrow[3]{} C_2 \xleftrightarrow[4]{} C_3 \xleftrightarrow[2]{} C_4$ min is 2

but we want max

0 3 4 7 9 10

$C_1 \xleftrightarrow[3]{} C_2 \xleftrightarrow[4]{} C_3 \xleftrightarrow[3]{} C_4$ min is 3

but we still increase to find min

0 3 4 7 9 10

$C_1 \xleftrightarrow{4} C_2 \xleftrightarrow{5} C_3 \longleftrightarrow$

not possible

So answer is 3

suppose take

arr[] = [0 3 4 7 9 10]   cows=2

to get max(min)

U will place 1 cow in 1st stall

and another cow in last stall

$$c_1 \xleftrightarrow{10} c_2$$

[0 3 4 7 9 10]

function to check whether we can place the all cows //(such that we need to get max) imp

```
boolean
function    checkplacecows (arr, dist, cows)
{
    int countcows=1;
    last = arr[0];
    for(int i=1;i<arr.length;i++)
    {
          if (~~dists~~ arr[i]-last>=dist)
                         ~~continue~~
          {
           ~~else~~
               countcows++;
               last= arr[i]
          }
    }
    if(countcows>=cows)
                 return true

    else
          return false;
}
```

```
public  int   solve (int[] a, int cows)
{
        int low=1;              max in array a
        int high= max-min       min in array a

        while (low<=high){
        {
                int mid= low+(high-low)/2

                if (canweplaceCows(a,mid,cows))
                {
                        low=mid+1;
                }
                else{
                        high=mid-1;
                }
        }
        return low;
}
```

```
     ✓   ✓   ✓   ✗   ✗   ✗   ✗   ✗   ✗   ✗
low  1   2   3   4   5   6   7   8   9   10  high
            high low
```

at Last high reaches the possible one