

Approach

Question

Given an array of length 'N', where each element denotes the position of a stall. Now you have 'N' stalls and an integer 'K' which denotes the number of cows that are aggressive. To prevent the cows from hurting each other, you need to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. Return the largest minimum distance.

Eg

array: 1,2,4,8,9 & k=3

O/P: 3

Explanation: 1st cow at stall 1 , 2nd cow at stall 4 and 3rd cow at stall 8



Logic:

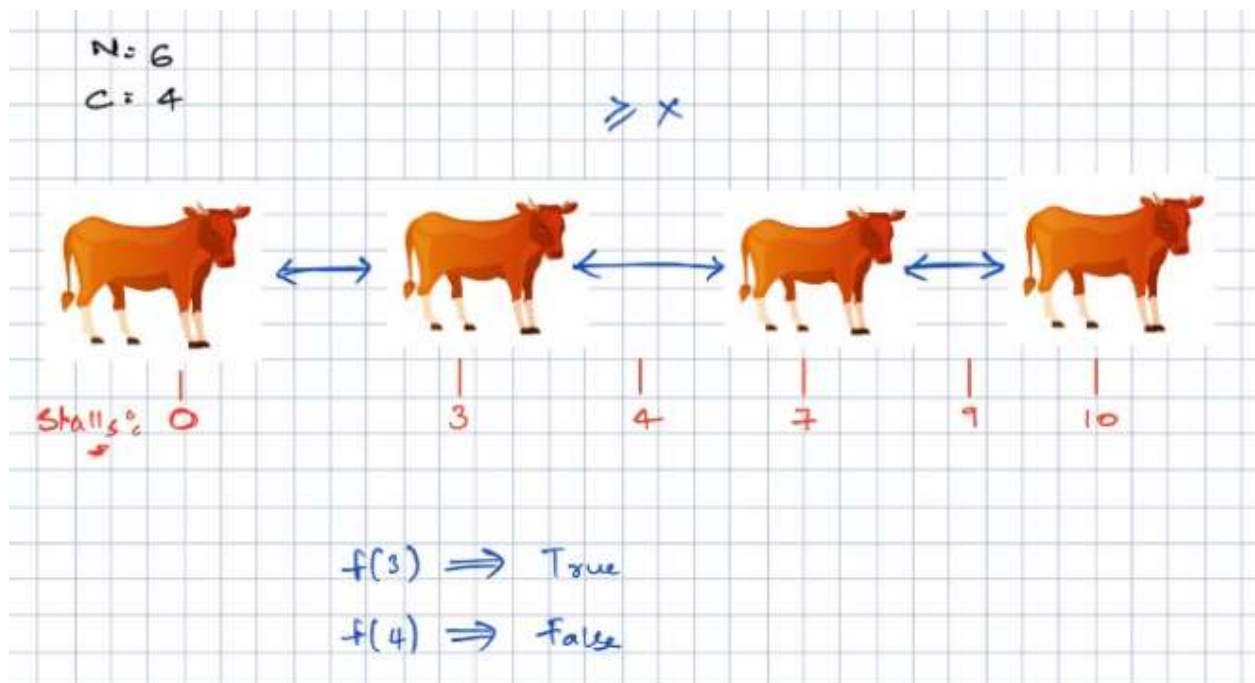
- We need to define a `isPossible()` function that checks if a distance `x` is possible between each of the cows. We can use a **greedy** approach here by placing cows at the **leftmost** possible stalls such that they are at least `x` distance away from the last-placed cow.
- We need to sort the given array/list so once we have our sorted array to do the greedy task then we can apply the **Binary Search** algorithm on the sorted input, and use our function `isPossible()` to find the largest distance possible.

Steps are as follows :

- First of all, consider low and high values for the starting and ending positions of the given array/list or just take low=0 and high = ending position (anything it really does not matter ...you can even take low=0 and high=100000 any large no ...eventually binary search will eliminate the extra range by removing their left half or right half)
- Let mid be the minimum distance. Check in the sorted array how many positions are available so that the minimum distance between them is this mid.

Now how to do that?

1. Start with the least value in the array and keep ignoring all the elements until you get a difference between the lower element and the current element \geq mid or you run out of elements.
 2. Use this idea to find the count of such possible positions. Now if this count is $\geq c$ then we already have a possible solution so store it and we now eliminate it from our search space by making low as mid+1, otherwise, we don't store the result as it is not our solution and eliminate it from our search space by making high as mid-1.
- . Now our only objective is to find the maximum from all the results we have stored until our search space has exhausted.
 -



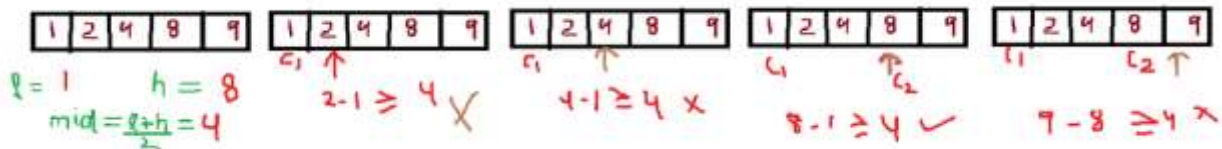
Basically :

- If you placed all cows with min dist 'mid' this means we can try to further increase this min dist so we do low=mid+1

- But if all the cows could not be placed within the 'mid' dist WKT that increasing the dist further would not be of use since we could not ensure even the min distance of 'mid' b/w cows ...how can we think of a greater distance than 'mid' ...SO We know for sure that the dist has to be less than mid so we do $high = mid - 1$

Suppose we take 1,2,4,8,9 and cows=3 (given)

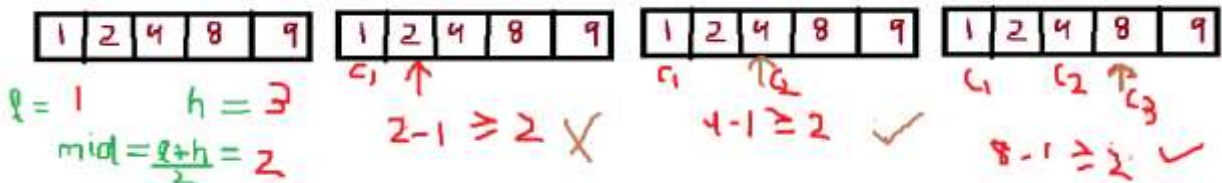
So initially $low = 1$, $high = 8$ and therefore $mid = (1+8)/2 = 4$



We could only place 2 cows not 3 so return false as '4' dist b/w cows is not possible it has to be less than that

So we do $high = mid - 1 = 4 - 1 = 3$

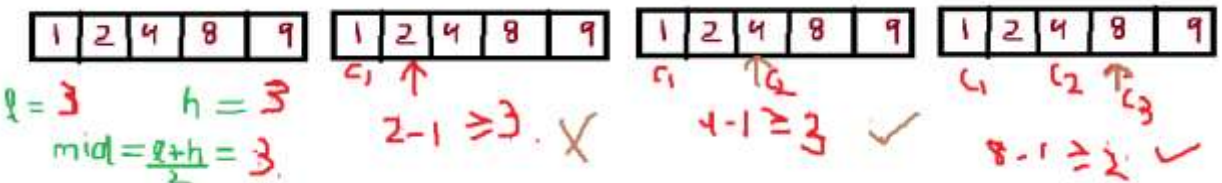
so now $low = 1$, $high = 3$ and so mid becomes average of 1 & 3 i.e 2



Now we know one possible ans is 2 ..since we need to maximise this min ans so lets try for a dist > 2

so we do $low = mid + 1$ i.e $2 + 1 = 3$

so $low = 3$ & $high = 3$ so mid becomes 3



Now we got 3 as ans WOW lets try for more dist

$low = mid + 1$ i.e $low = 3 + 1 = 4$

$low = 4$ and $high = 3$ OOPS low crosses high so STOP

3 is the only ans !!