

AGGRESSIVE COWS

☆ We are given an array of integers where each integer will represent coordinate of a stall. We are supposed to place cows in these stalls such that the minimum distance between any cows in a specific configuration is maximum amongst all specific configuration.

Distance in a specific configuration will only be minimum when measured between two adjacent elements.

We have to sort the array.

Brute force solution is to iterate through natural numbers from 1 to $(\max - \min)$, and check that if by maintaining the minimum distance, we can fit all cows, that answer is possible. Since the array is sorted, first cow will always be on the first index.

Optimal solution is to make this search happen with Binary Search

Pseudocode :

```
placementPossible(arr, N, d, c) {  
    k = 1, cu = 0  
    for(i → 1 → N - 1) {
```

```
if (arr[i] - arr[cu] >= d) {  
    k++  
    cu = i  
}
```

```
}  
if (k >= c) {  
    return true  
} else {  
    return false  
}
```

```
}  
aggressiveCows(arr, N, c) {
```

```
    min, max = 0, 0
```

```
    for (i → 1 → N - 1) {
```

```
        if (arr[i] < min) {
```

```
            min = arr[i]
```

```
        }
```

```
        if (arr[i] > max) {
```

```
            max = arr[i]
```

```
        }
```

```
    }
```

```
    high = max - min, ans = -1
```

```
    low = 1
```

```
    while (low <= high) {
```

```
        mid = (low + high) / 2
```

```
        ret = placementPossible(arr, N, mid, c)
```

```
        if (ret) {
```

```
            low = mid + 1
```

```
        } else {
```

```
            high = mid - 1
```

```
        }
```

```
    }
```

} return ans