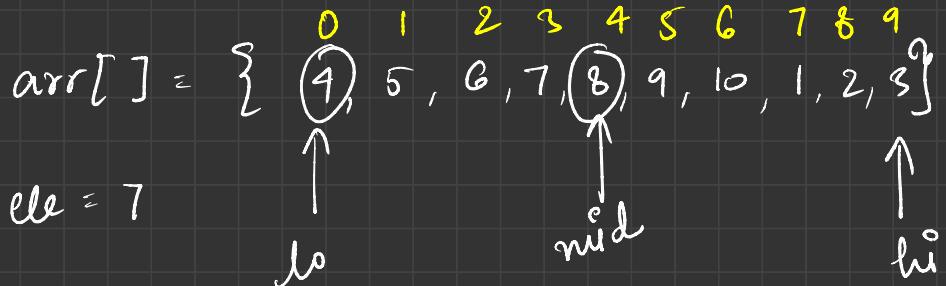




Search In A Rotated Array

$\text{arr}[0] \leq \text{arr}[\text{mid}]$

$\xrightarrow{\quad\quad\quad}$ left Part Sorted $\xrightarrow{\quad\quad\quad}$



{

if ($\text{arr}[\text{mid}] = \text{ele}$)
return mid;

else if ($\text{arr}[0] \leq \text{arr}[\text{mid}]$) \rightsquigarrow left Part is sorted.

 if (ele In Range)
 move left;
 else
 move right

else \rightsquigarrow Right Part Sorted.

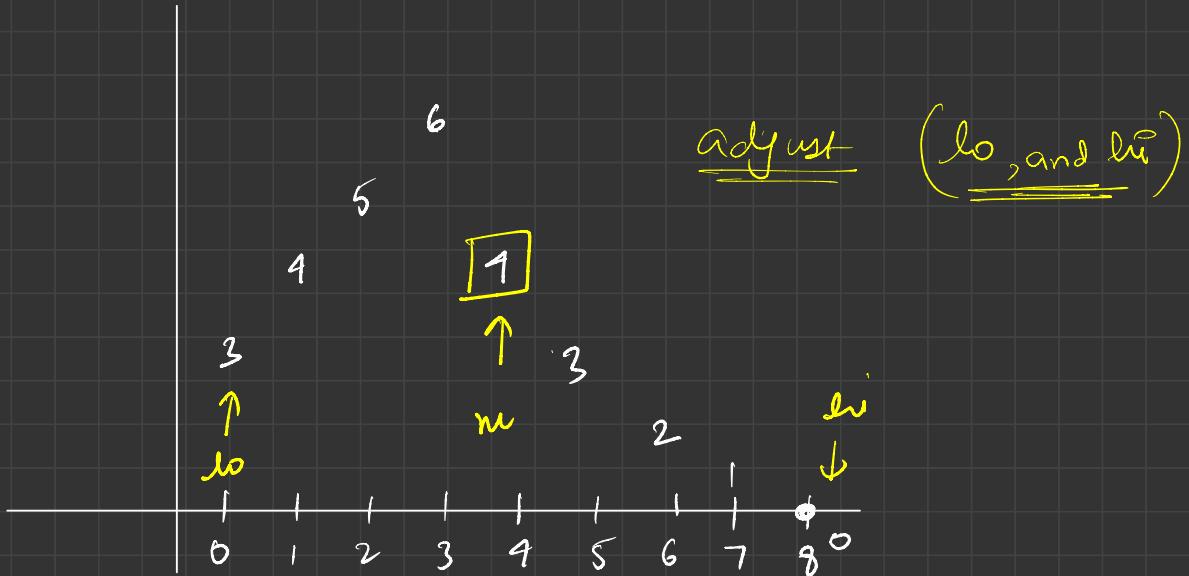
{
 if (ele in Range)
 move Right;
 else
 move Left; }
}

arr[] = { 0 1 2 3 4 5 6 7 8 9 }
 { 9, 5, 6, 7, 8, 9, 10, 1, 2, 3 }

ele = 2

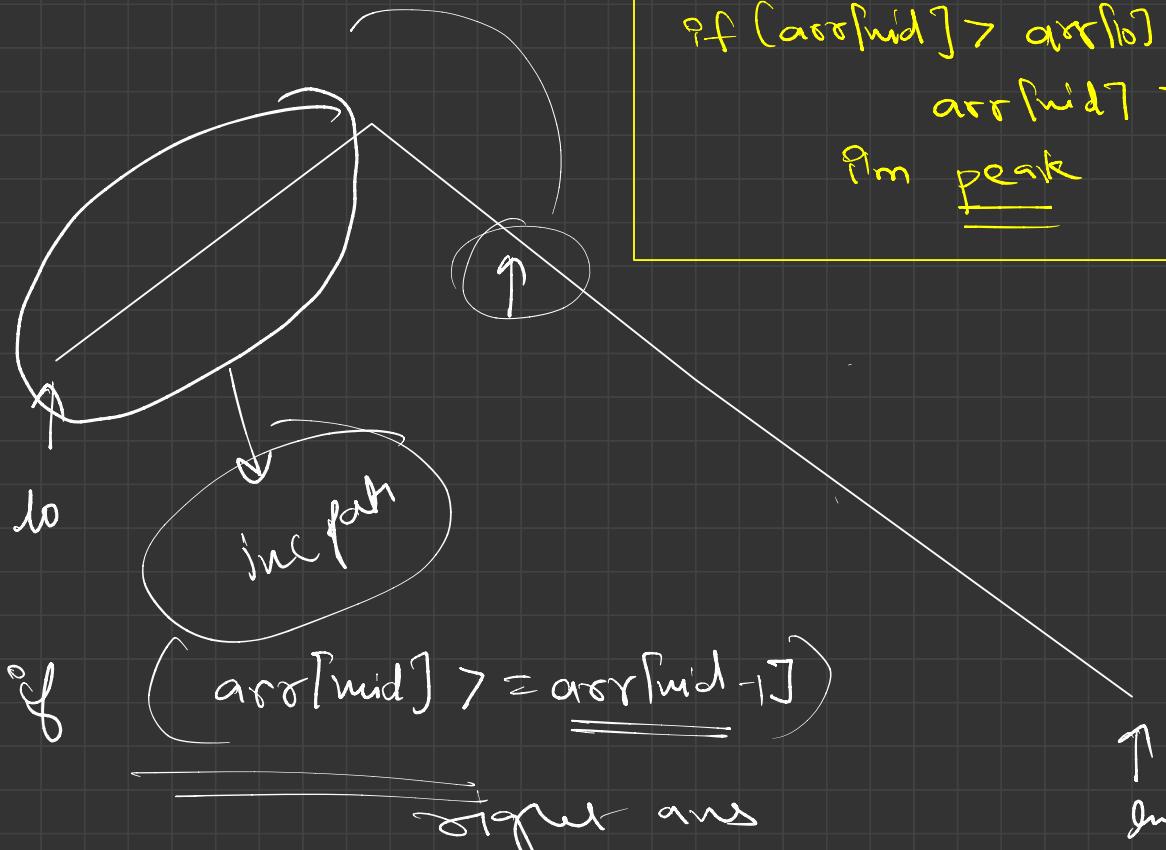
peak in a Mountain Array

→
[3, 9, 5, 6, 4, 3, 2, 1, 0]



if ($\text{arr}[\text{mid}] > \text{arr}[\text{lo}]$ and
 $\text{arr}[\text{mid}] > \text{arr}[\text{hi}]$)

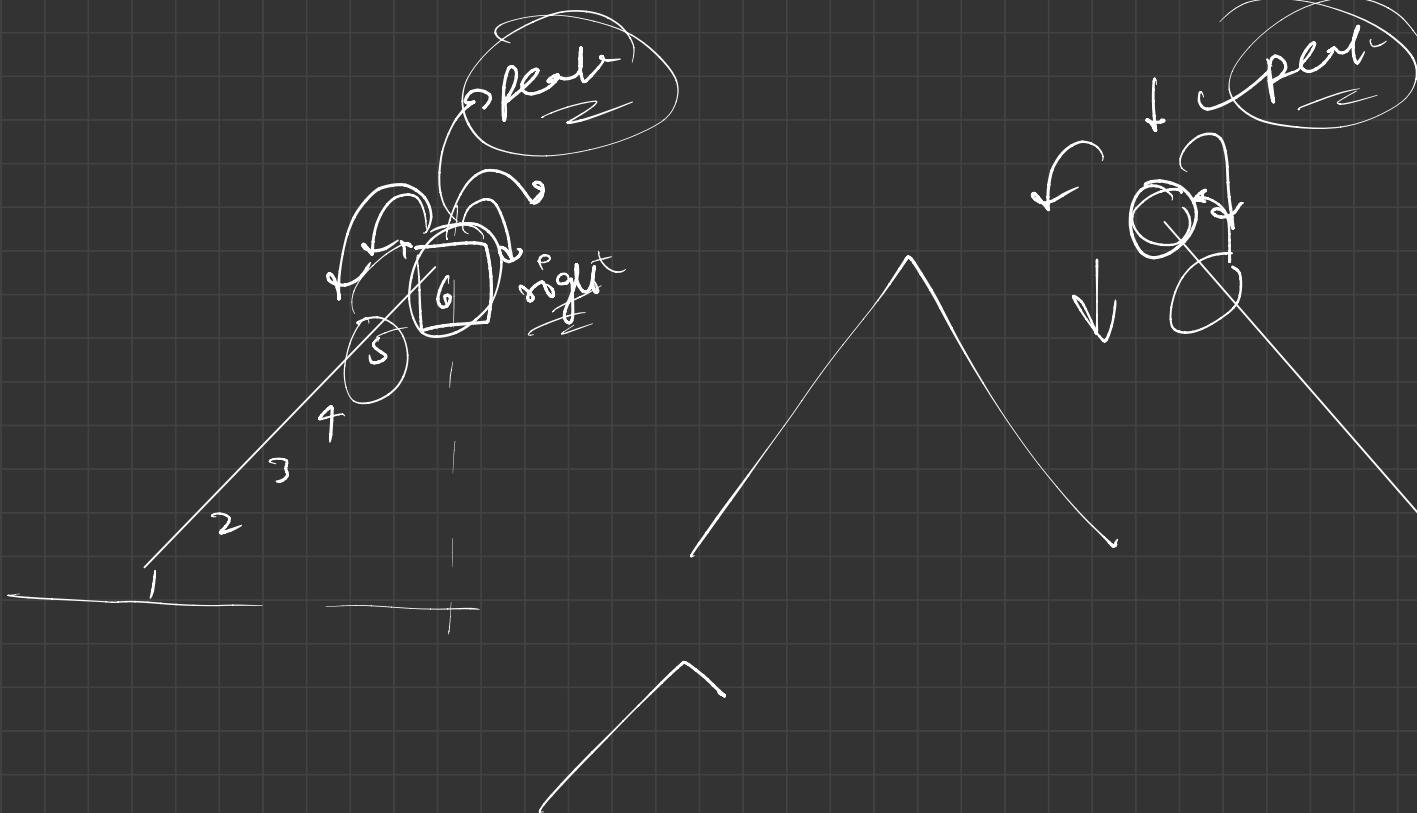
in peak



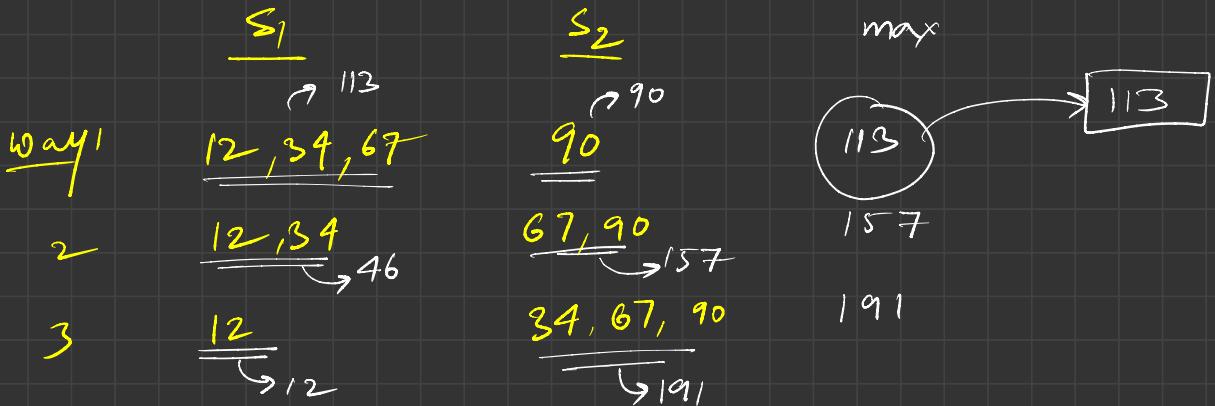
if ($\underline{\text{arr}[\text{mid}] \geq \text{arr}[\text{mid}-1]}$)

right ans

else left ans



$\text{books}[] = [12, 34, 67, 90]$ stud = 2



$\text{books} \}] = \{ 12, 34, 67, 90 \}$ Stud = 2

Stud = 1



$$12 + 34 + 67 + 90$$

- 203 pages

98 stud



90 pages

Stud = 4

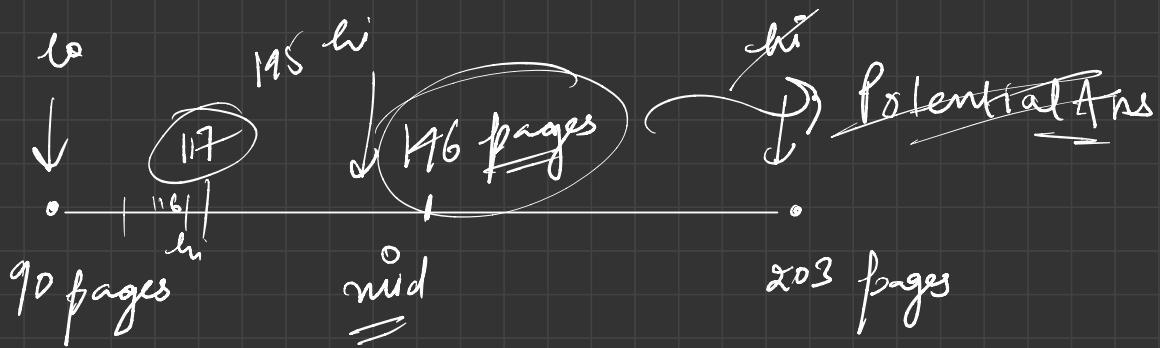
12 34 67 90 ~ 90

= 90 pages

1 stud



203 pages



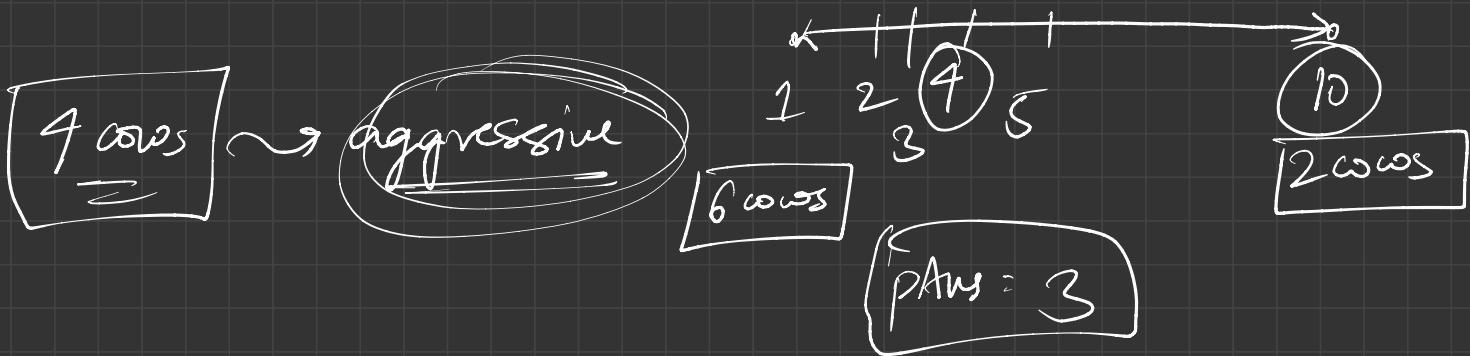
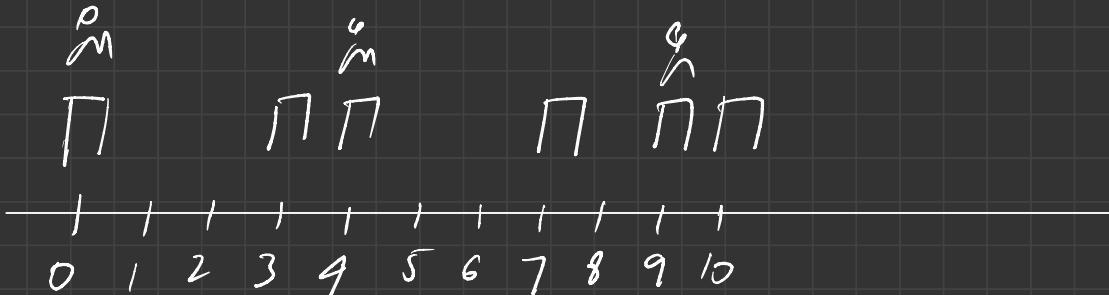
$\text{books}[] = \{ 12, 34, 67, 90 \}$

$$s_1 = 12 + 34 + 67$$

$$s_2 = 90$$

Aggressive Cows

stalls $\] = \{ 0, 3, 1, 7, 10, 9 \}$



```

static boolean isPossibleToPlace(int[] a, int cows, int capDist) {
    int lastPositionPlaced = 0;
    int numberOfCowsPlaced = 1;

    for (int i = 1; i < a.length; i++) {
        int distBetweenNewStallAndPrevStall = a[i] - a[lastPositionPlaced];
        if (distBetweenNewStallAndPrevStall >= capDist) {
            numberOfCowsPlaced++;
            lastPositionPlaced = i;
        }
    }

    return numberOfCowsPlaced >= cows;
}

// TC: O(N * log N), SC: O(1)
public static int aggressiveCows(int a[], int cows) {
    // Write your code here
    Arrays.sort(a);

    int maxDist = a[a.length - 1] - a[0];
    int minDist = 1;

    int lo = minDist;
    int hi = maxDist;
    int ans = 0;
    while (lo <= hi) {
        int capDist = (lo + hi) / 2;

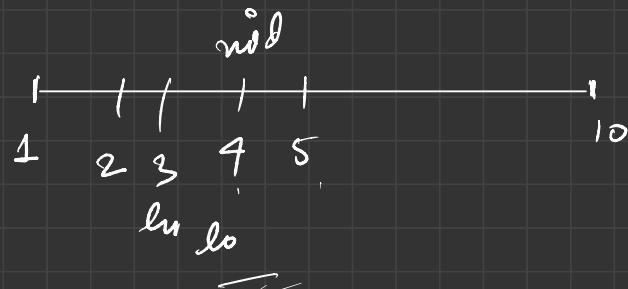
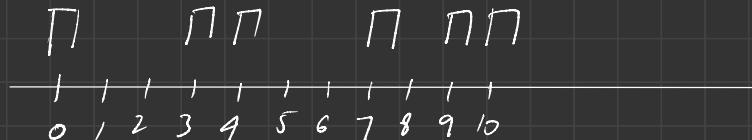
        if (isPossibleToPlace(a, cows, capDist) == true) {
            ans = capDist;
            lo = capDist + 1;
        } else {
            hi = capDist - 1;
        }
    }

    return ans;
}

```

$$\text{stalls}[] = \{ \begin{matrix} \textcircled{1}, & \textcircled{X}, & \textcircled{1}, & \textcircled{X}, & \textcircled{1}, & \textcircled{X} \\ 0, & 3, & 7, & 9, & 10 \end{matrix} \}$$

$$\text{cows} = 4$$



ans = 2 (3)

