



Agenda

1. Search in a 2D Matrix
2. min in rotated sorted array
3. find in rotated sorted array
4. peak index in a mountain array
5. Allocate min. no of pages

Search in
2D Sorted Array

clear!

	0	1	2	3
0	1, 2, 3, 4			
1		5, 6, 7, 8		
2		9, 10, 11, 12		
3		13, 14, 15, 16		
4		17, 18, 19, 20		

ele = 15

① Each row will be sorted

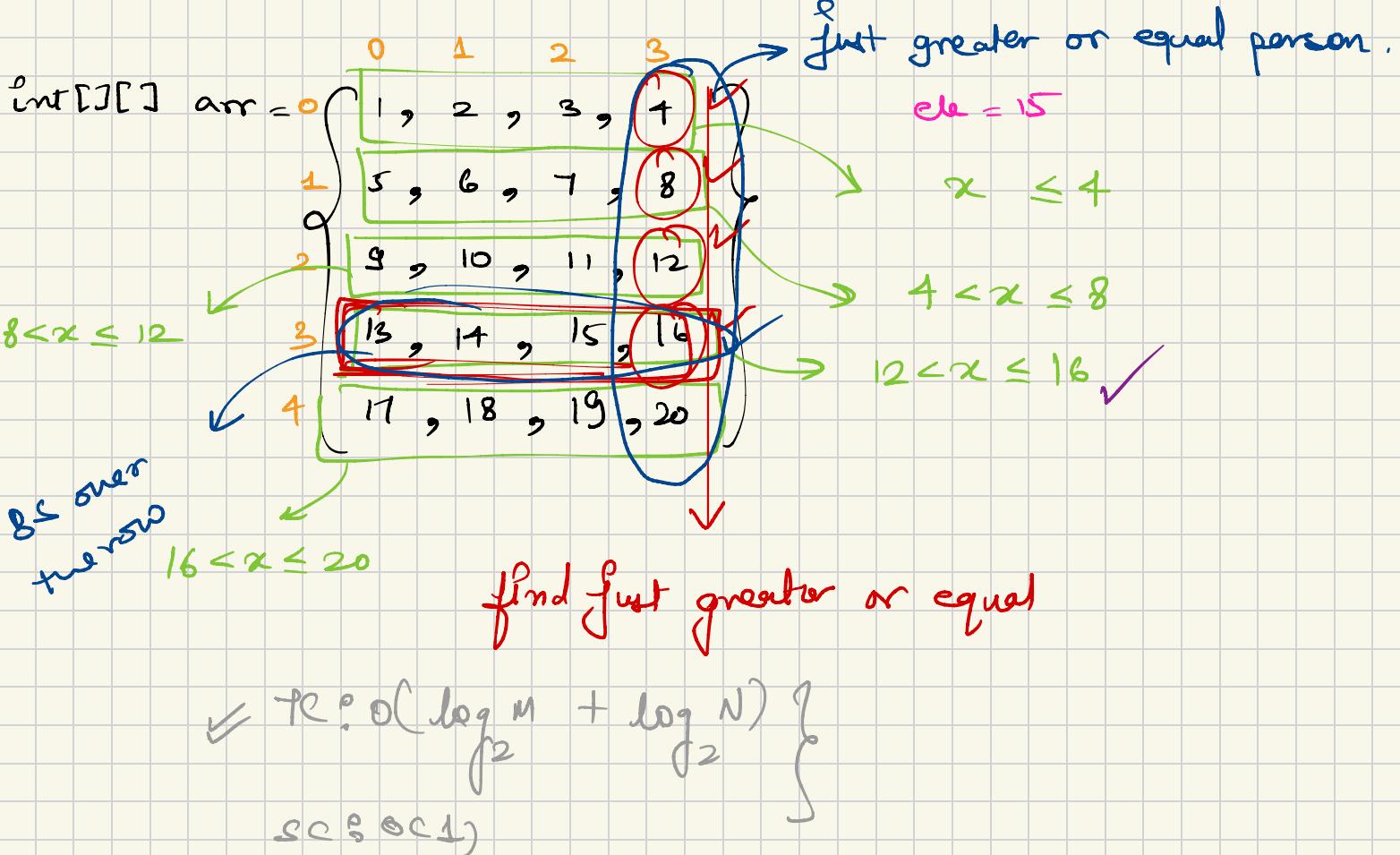
② last ele of the prev row, will be smaller than
first ele of next row

Brute Force,

W.F. $O(N^2)$
W.A. $O(N^2)$

TC: $O(N \times M)$
SC: $O(1)$

```
for (int i = 0 → n)
{
    for (int j = 0 → m)
        if (arr[i][j] == target)
            return true;
}
return false;
```



find min^m in a rotated sorted array / pivot index

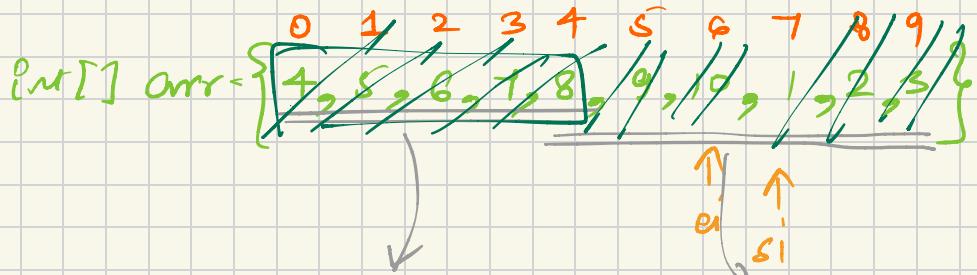
int[] arr = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
 {4, 5, 6, 7, 8, 9, 10, 1, 2, 3}

Brute force

o Linear Search

→ find min ✓

↑
 $T = O(N)$ & $S = O(1)$



Binary Search

prin = ~~1~~ 1

part1
 ↓
 Sorted part

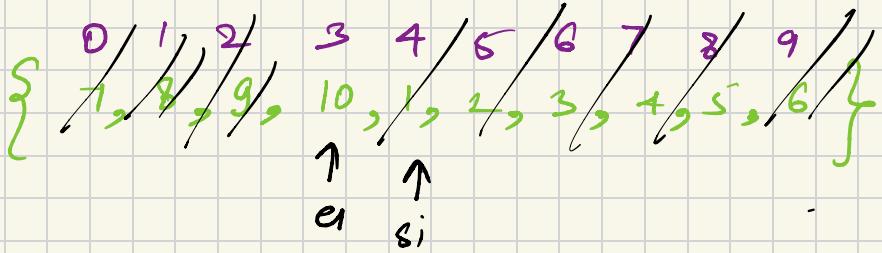
part2
 ↑
 mid unsorted part

4 \leq a \leq 8

Sorted part

✓
 arr[fp] \leq arr[lp]

- ✓
- {
- o split the array in 2 parts
 - o store min from sorted region,
 - o find in another region

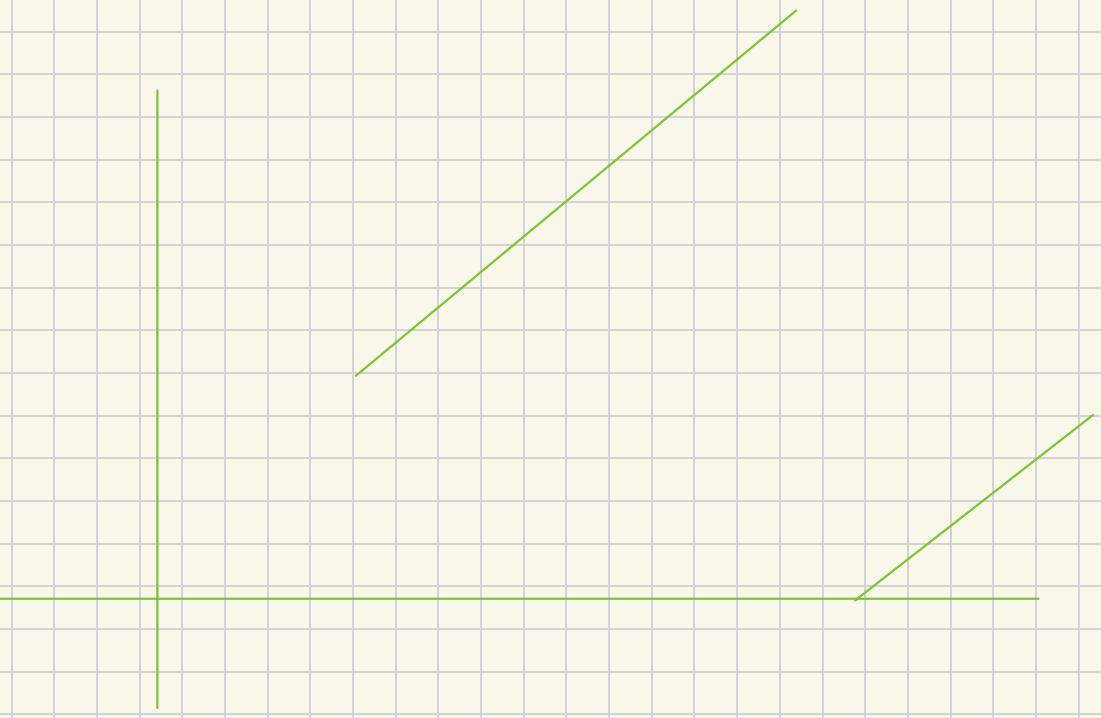


clear?

$$\checkmark_{\text{pwhr}} = 1$$

1
no 4

\checkmark TC: $O(\log N)$ SC: $O(1)$



Find element in a rotated sorted array.

$$\text{arr} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8 \\ 5, 6, 7, 8, 9, 10, 2, 3, 4 \} \quad \text{ele} = 3$$

Approach 1

- get pivot
- apply BS in sorted which might ignore case

$$TC: O(\log N) + O(\log_2 N)$$

$$= \underline{\underline{O(\log_2 N)}}$$

$\text{arr} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8 \}$
 $\underline{\underline{5, 6, 7, 8, 9}}, 10, 2, 3, 4 \}$ $\text{ele} = 3$
 ↑ ↑
 si ei
 ↑
 mid

Case 1 $\text{arr}[\text{mid}] == \text{ele}$

Case 2 left is sorted

- in range → move left
- not in range → move right

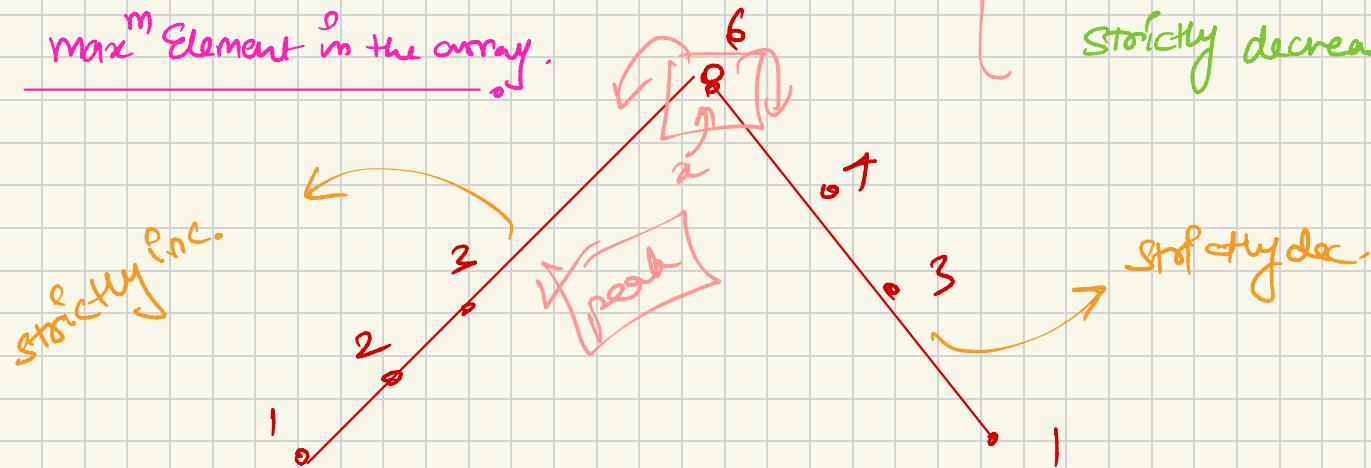
Case 3 right is sorted

- in range → move right
- not in range → move left

peak element in a mountain array

max^m Element in the array.

clear
strictly inc,
strictly decreasing

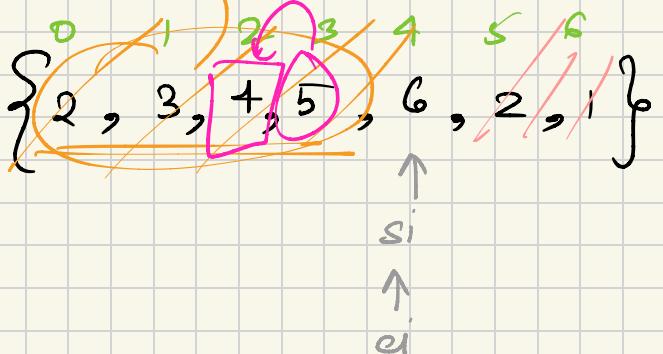


x^m is peak

if ($\text{arr}[x-1] < \text{arr}[x]$ & $\text{arr}[x] < \text{arr}[x+1]$) ✓

left side

arr



Case 1 mud pink ↑
mud

Case 2 left inc, arr[mud-1] < arr[mud] :

move right

Case 3. left not inc → move left

$\lceil \text{TC} : O(\log N) \text{ sc}: O(1) \rceil_2$

s
2
3

4

