



## Search in 2D Matrix

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

key = 14

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

SC:  $O(1)$

- smaller or equal to last person of the row.
- greater than last person of the prev. row

just greater or equal element.

to find row      to find key

## Binary Search { Searching Algo }

divide search region, into 2-half and take one and eliminate another

$$Tc: O(\log_2 N) \quad sc: O(1)$$

## Agenda

- find pivot in a rotated sorted array
- find element in a rotated sorted array
- peak index in a mountain array
- peak index
- Binary Search over  $201^n$

## Rotated Sorted Array

✓  $\text{int[] arr} = \{4, 5, 6, 7, 8, 9, 1, 2, 3\}$

first element to start rotation {pivot}

↓  
smallest element in the array

Search Pivot in a rotated sorted array.

int[] arr = { <sup>0</sup>~~4~~, <sup>1</sup>~~5~~, <sup>2</sup>~~6~~, <sup>3</sup>~~7~~, <sup>4</sup>~~8~~, <sup>5</sup>9, <sup>6</sup>1, <sup>7</sup>2, <sup>8</sup>3 }

Sorted

unsorted

min = ~~4~~ ~~1~~ 1

↑  
ei  
↑  
mid

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

$\{ 5, 6, 7, 8, 9, 1, 2, 3, 4 \}$

$\uparrow$   $\uparrow$   
 $ei$   $si$

$minVal = \cancel{4} 1$

$\uparrow$   
 $mid$

arr = { ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~, 9, 1, 2, 3 }

$\uparrow$   $\uparrow$   $\uparrow$   
si mid ei

if  $arr[x-1] > arr[n]$  : pivot at  $x^{th}$  index

if  $arr[y] > arr[y+1]$  : pivot at  $y+1$ th index



find element in a rotated sorted array.

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

Indices: 0 1 2 3 4 5 6 7 8

Annotations:

- A pink line is drawn from index 0 to index 1.
- A pink box highlights the subarray [1, 2, 3] at indices 6, 7, and 8.
- Orange arrows point to indices 5, 6, and 7, labeled  $si$ ,  $mid$ , and  $ei$  respectively.

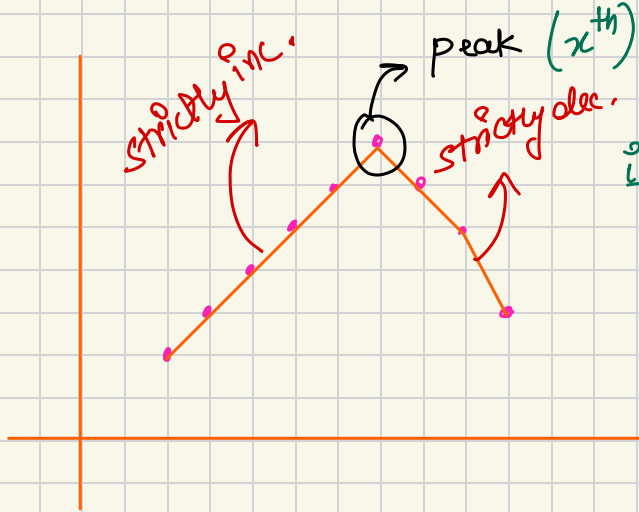
key = 2

→ [1, 8]

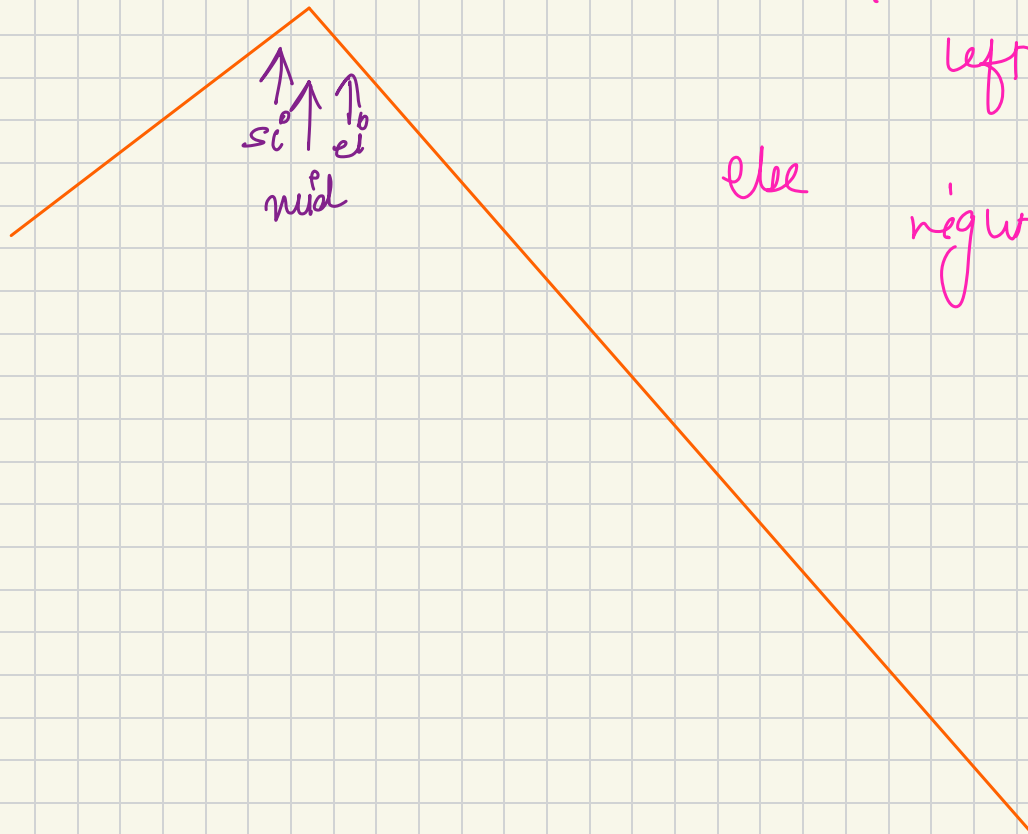
[1, 3]

## Peak in a Mountain array.

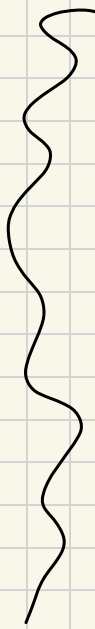
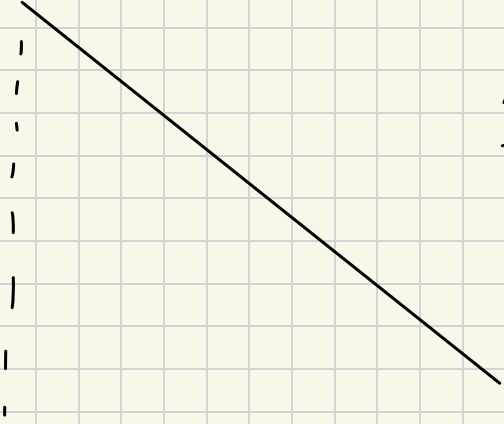
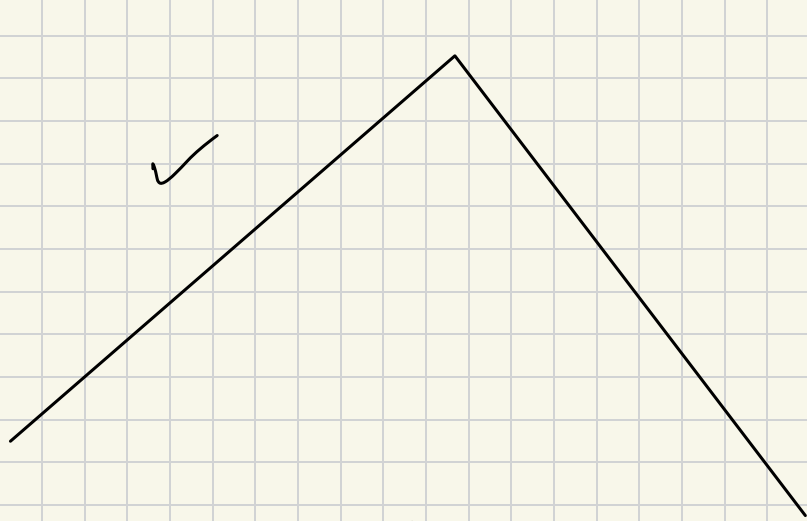
$\text{int[]} \text{arr} = \{ 1, 2, 3, 4, 5, 6, 5, 4, 2 \}$



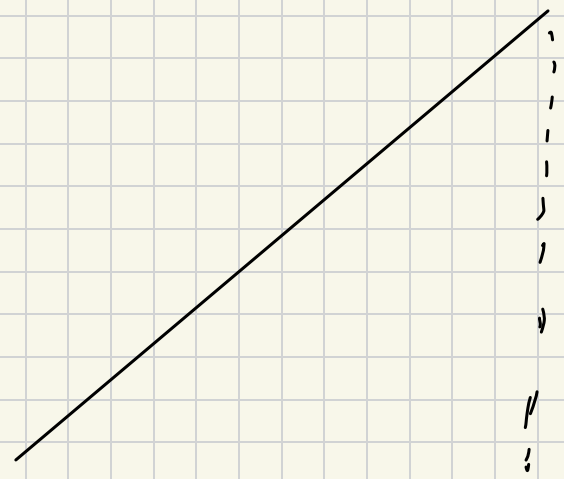
if  $(\text{arr}[n+1] < \text{arr}[n] \ \&\& \ \text{arr}[n-1] < \text{arr}[n])$

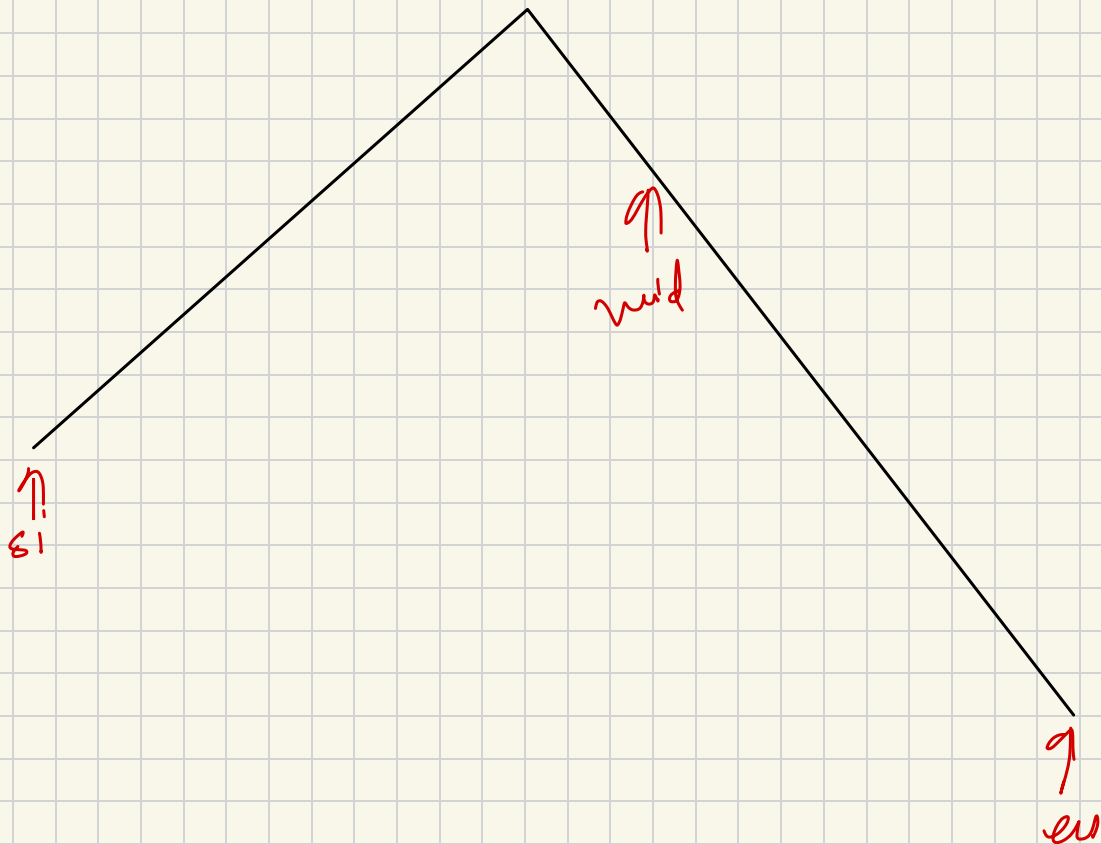


if ( $\text{arr}[\text{mid}] - 1 \leq \text{arr}[\text{mid}]$ )  
left inc.  
else  
right is dec



take care  
special cases





int[] = {<sup>0</sup>4, <sup>1</sup>5, <sup>2</sup>6, <sup>3</sup>7, <sup>4</sup>8, <sup>5</sup>9, <sup>6</sup>7, <sup>7</sup>5, <sup>8</sup>3}

↑  
s  
↑  
e  
↑  
mid

Allocate Min<sup>m</sup> No. of Pages

int[] Books = { 24, 12, 67, 90 }      students = 2

- distribute these N Books among M students
- each student gets min one Book
- distribution should be contiguous

int[] Books = { 34, 12, 67, 90 }      students = 2

way 1

s1 34, 12, 67 → 113 pages  
s2 90 → 90 pages

way 2

s1 34, 12 → 46 pages  
s2 67, 90 → 157 pages

way 3

s1 34 → 34 pages  
s2 12, 67, 90 → 169 pages

Max<sup>m</sup> By any stud.

113 pages

157 pages

169 pages

min 113 pages

min. of max. pages read by a stud.