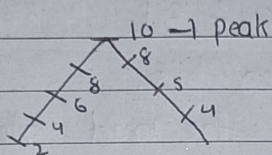


Day - 32Question on Binary Search - 2\* Peak Index in Mountain Array:

0	1	2	3	4	5	6	7
2	4	6	8	10	8	5	4

=> So here, we have to return the index of peak element.

=> 1st Approach

=> By using LS, store the index of largest element and after iterating the array return the index.

=> Time Complexity:  $O(N)$

=> 2nd Approach

=> By using BS, we can know that the peak element is greater than their both left & right element.

=> So, we use this for finding the peak element.

=> For finding <sup>correct</sup> mid, when the ~~m~~ element left of mid is smaller, you have to go to right direction otherwise left.



\*

### Rotated array:

=>

2	4	6	8	10	
0	1	2	3	4	5
4	6	8	10	1	2

=>

Again we use BS for this.

=>

Now, here, we will find the mid element about it is left sorted or right sorted. Then ~~it~~ update the start and end values according to it.

=>

For this, we have to check the mid element with first element.

=>

If it is greater than first element then it is left sorted otherwise right sorted.

=>

Left side sorted  $\rightarrow$  Right side  
Right side sorted  $\rightarrow$  left side

=>

Also, we store the index as answer when we go to left side.

\*

### Search in Rotated Array:

target = 1

4	5	6	7	0	1	2
0	1	2	3	4	5	6

=>

We use same concept, we do in above question, but instead of checking with arr[0] here we check with target.



Date \_\_\_\_\_

Page \_\_\_\_\_

⇒ In this, first we check with array for left & right sorted part then check the target element lies in that sorted part or not and take actions according to it.