



Binary Search

- ① define a range (region of search)
- ② try to eliminate half of the region
- ③ and continue search in the other half.

$$\begin{cases} \text{TC: } O(\log_2 N) \\ \text{SC: } O(1) \end{cases} \rightarrow \text{length of search region}$$

Q
Ans

Search minimum value in a Rotated Sorted Array.
Search pivot index in a Rotated Sorted Array.

arr[] = { 20, 40, 45, 46, 70, 1, 2, 7 }

0 1 2 3 4 5 6 7

rotated Sorted array

pivot

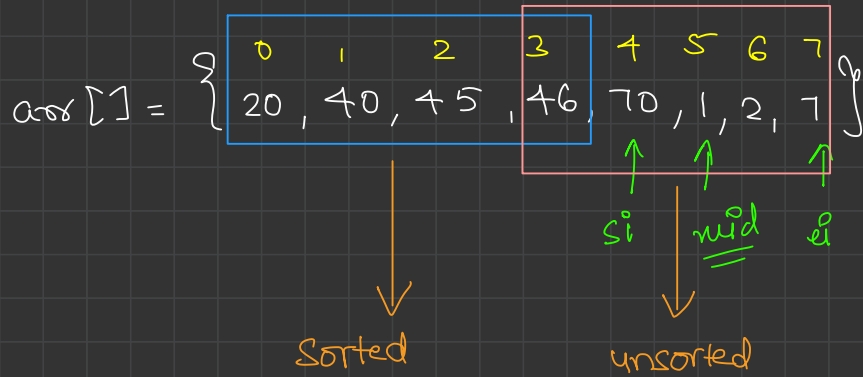
Brute force

↳ Linear Search

find min^m value's index

TC: $O(N)$

SC: $O(1)$



mid will divide array into sorted and unsorted region, and pivot always lies in unsorted region!

if (arr[si] <= arr[mid])
left side sorted

else if (arr[mid] <= arr[ei])
right side sorted.

~~if~~
if (arr[mid-1] > arr[mid])
mid → pivot

Search in a rotated sorted array

arr = {⁰20, ¹40, ²45, ³46, ⁴70, ⁵1, ⁶2, ⁷7}

target = 2

sorted

[20, 46]

pivot

si

wid

ei

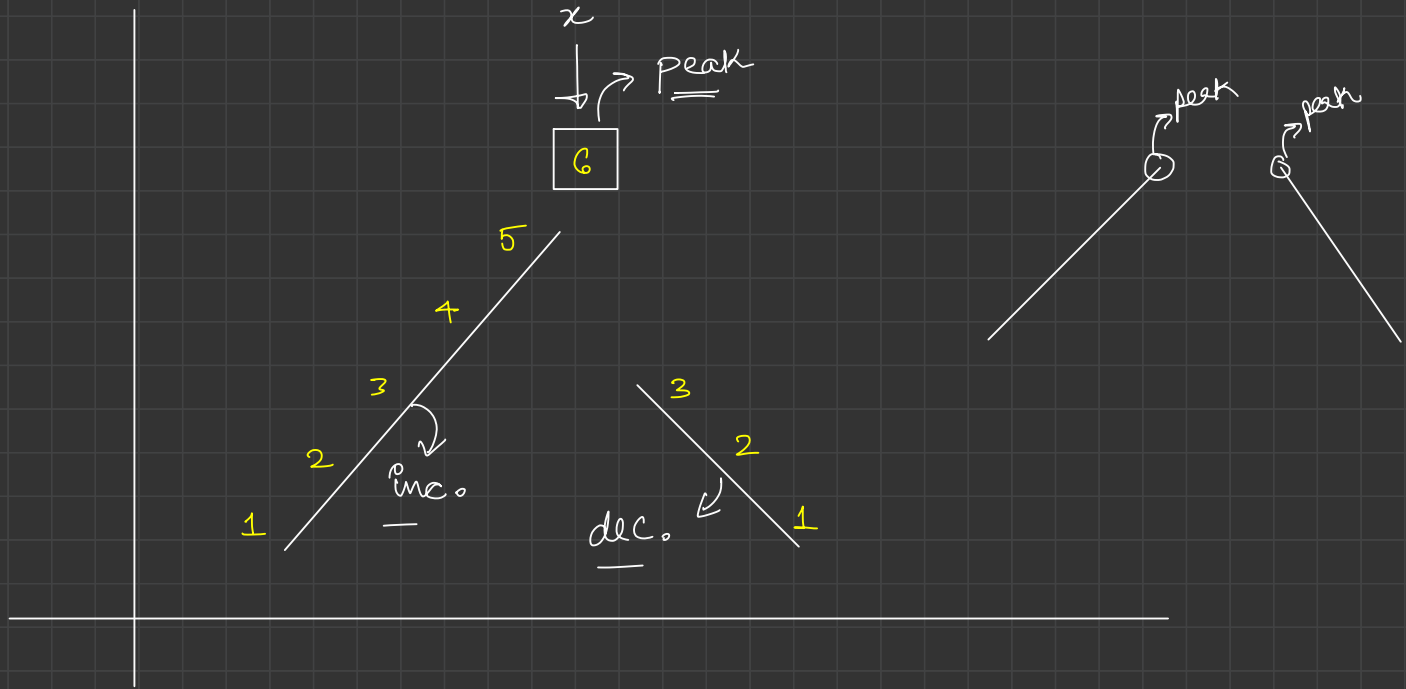
[1, 7]

Brute force

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

Peak Index in a Mountain Array

[1, 2, 3, 4, 5, 6, 3, 2, 1]



for x to be a peak

$$\text{arr}[x-1] < \text{arr}[x] \text{ \&\& } \text{arr}[x+1] < \text{arr}[x]$$

Case 1

