

Binary Tree.

"श्री कृष्ण गोविंद हरे मुरारी, हे नाथ नारायण वासुदेवा"

## Data Structure and Algorithm

By

Malay Tripathi

\* FIND MINIMUM IN ROTATED SORTED ARRAY, [UNIQUE].

arr[] = [4 5 6 7 0 1 2]

[0 1 2 4 5 6 7]

← Ideally Sorted Array like to be this: →

→ So, minimum is "0", in this we have to return values.

Brute Force.

$O(n) = T.C.$

Step 1:- always eliminated either left hand side or right hand side of the array.

Step 2: Identify the sorted half.

arr[] = [4 5 6 7 | 0 1 2]

l

m

h

→ Left hand side is sorted because  $4 \leq 7$

→ right hand side is not sorted.  $7 \leq 2$  (Ideally value at mid should be less than high).

→ Because of rotation → Right half is not sorted.

→ ONE THING WE CAN SAY THAT, THE VALUE FROM WHICH VALUE IS ROTATED IS ALWAYS THE MINIMUM VALUE.

→ We can say sorted half may or may not have an answer.

arr[] = [4 5 6 7 | 0 1 2]

sorted,

x

m

h

left half is sorted. -

ans = INT-MIN ~~40~~

[0] Ans.

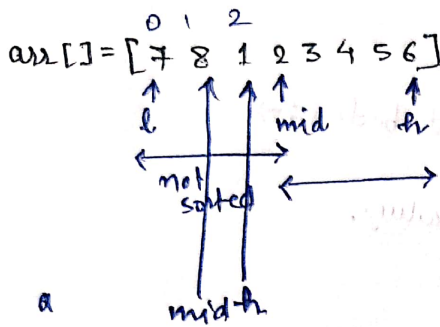


“श्री कृष्ण गोविंद हरे मुरारी, हे नाथ नारायण वासुदेवा”

## Data Structure and Algorithm

By

Malay Tripathi



ans = INT-MAX

ans = 2

so, left half is sorted,

arr[] = [7 8 1 2 3 4 5 6]

↑

h

l

m

ans = 1.

Ex 3 → arr[] = [4 5 1 2 3]

↑ ↑ ↑

low m high

ans = INT-MAX

l.h is not sorted  $4 < 1$  | right hand is sorted.  $1 < 3$ .

ans = INT-MAX 1

arr[] = [4 5 1 2 3]

h

l

m

ans = INT-MAX 1.

0 4 is lowest in left half but not used.

“श्री कृष्ण गोविंद हरे मुरारी, हे नाथ नारायण वासुदेवा”

## Data Structure and Algorithm

By

Malay Tripathi

f(arr, n)

{  
low=0, high=n-1, ans=INT-MAX

while(low <= high)

{  
mid = (low+high)/2

if (a[low] <= a[mid])

{

ans = min(ans, arr[low])

low = mid+1;

}

else

{

ans = min(ans, arr[mid])

high = mid-1;

}

}

return ans;

}