LEETCODE - (33) Search in Rotated Sorted Array.

→ ascending order; distinct values. (Modified BS).

TC : O(logn).

[3, 4, 5, 6, 7, 0, 1, 2] tar = 0.

Pseudocode :-

```
st=0, end = n-1
while( st <= end) {
    mid = st + (end-st)/2
    if (A[mid] == tar) → mid.
    if (A[st] <= A[mid])   // left sorted
        if (A[st] <= tar <= A[mid]) → left (end = mid)
        else → Right → st = mid+1.
    else // Right sorted
        if (A[mid] <= tar <= A[end]) → Right (st = mid+1)
        else → left → end = mid-1.
```

LS ↙ RS

① find mid,

② check which Half is sorted.

to do that, if st < mid → LEFT sorted

if end > mid → RIGHT sorted.

③ Apply BS to side which is sorted.

IF target lies in the [sorted half]

Left	Right
i.e. (st < tar < mid)	i.e. mid < tar < end
↓	↓
end = mid-1	st = mid+1.
else, target belongs in RIGHT	else, target belongs in LEFT.
so, st = mid+1	so, end = mid-1