Binary Search - Sorted Rotated Array :

Experiment :

Date

Page No.

Rotation :

→ 2, 3, 4, 7, 11

Rotation → Clock Wise ↻
→ Anti-Clock Wise ↺

(•) Clock Wise →

7, 11 | 2, 3, 4, 7, 11

↓

• This will form an Unsorted Array.

(•) Anti - Clock Wise →

2, 3, 4, 7, 11 | 2, 3

→ CW

R = Rotation Count

R 0 → 2, 3, 4, 5

R 1 → 5, 2, 3, 4

→ ACW

R 0 → 2, 3, 4, 5

R 3 → 5, 2, 3, 4

Both are same except Rotation Count

• The only difference is that both have different rotation count.

DELTA®

→ CW RA → RC → index of minimum element.
→ ACW → RC → (N - index) % N;

Example of Anti Clock Wise Sorted: 5, 2, 3, 4

Total length → 4 - 1 = 3.

- Min element in RSA - CW

{64, 2, 3, 4, 5, 6.}

→ There have 3 conditions

Sorted	Sorted
Unsorted	Sorted
Sorted	Unsorted.

- Max is that element in which

Next element is ↓ smaller.

- Min is that element in which

Previous element is ↑ greater.

→ Q: → Skip Search Space

arr[] = 64, 2, 3, 4, 5, 6.

Experiment :

Date _____

Page No. _____

- Move towards Unsorted Half.
- If both sides is sorted
  Then move to left.
- Target : less than previous element

Dry Run :

```
arr[] = 64, 2, 3, 4, 5;
        0  1  2  3  4
        ↓     ↓     ↓
        S    Mid    E
```

Sorted Side

Then we will find in

```
[64, 2]
 ↓   ↓
 S   E
```

- Check that part is sorted or unsorted.

-> arr[end] > arr[mid]
   L> Right Side is Sorted.

-> If right side is sorted then ans will lie on Left Side of Array.

- End = mid - 1

DELTA®

else

start = mid + 1;

→ Q:

Find the max Element.

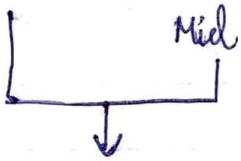
arr[] = {64, 9, 3, 4, 5}

0 1 2 3 4

64, 9, 3, 4, 5

↓

Mid



This area is not sorted. It means the answer lies here.

Conclusion:

If both are sorted

→ Move towards right.

Otherwise:

→ Move towards unsorted half.

- MAX Element:

→ The only element which is smaller than its next element.