⑤. Rotation Count in a Rotated Sorted Array

↓

find number of times the array was right-rotated from its originally sorted form.

↓

Note: array has distinct integers

IP:

0	1	2	3	4	5
15	18	2	3	6	12

OP: 2

arr[ ]:

0	1	2	3	4	5
15	18	2	3	6	12

sorted[ ]:

0	1	2	3	4	5
2	3	6	12	15	16

0	1	2	3	4	5	6	7
15	18	2	3	6	8	11	12

↑l   ↑mid   ↑r

$a[mid-1] < a[mid] < a[mid+1]$

0	1	2	3	4	5	6	7
6	8	11	12	15	18	2	3

↑l   ↑mid   ↑r

$a[mid-1] < a[mid] < a[mid+1]$

↓

only this cond^n can't decide where to go next

↳ look for the value a[0]

(a[0] > a[mid])

⇓

Go left

(a[0] < a[mid])

⇓

Go Right

0	1	2	3	4	5	6	7
15	18	2	3	6	8	11	12

↑l ↑mid ↑r

$a[mid-1] < a[mid] > a[mid+1]$

(ans = mid + 1)

0	1	2	3	4	5	6	7
6	8	11	12	15	18	2	3

↑l ↑mid ↑r

$a[mid-1] < a[mid] > a[mid+1]$

(ans = mid + 1)

edge case

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

↑l ↓mid ↑r

(mid = 0) ⟹ $a[0] < a[1]$ ⟹ ans = 0
            $a[0] > a[1]$ ⟹ ans = 1

0	1	2	3	4	5
6	7	10	11	13	15

↑l↑mid↑r

(mid = n-1) ⟹ $a[mid-1] < a[mid]$ ⟹ ans = 0
              $a[mid-1] > a[mid]$ ⟹ ans = n-1

0	1	2	3
32	18	26	30

↑l ↑mid ↑r

$a[mid-1] > a[mid] < a[mid+1]$ ⟹ (ans = mid)