Find Minimum in Rotated Sorted Array.

: Suppose an array of length n sorted in ascending order is rotated between 1 and n times. Return the minimum element of this array.

Your algorithm must run in O(log n) time.

↳ use binary search algorithm.

Ex)

4	5	6	7	1	2	3

Idea: 1. check if nums[0] < nums[-1].

↳ Yes: Not rotated, return nums[0]

2. Find mid located element.

0	1	2	3 (mid)	4	5	6
2	3	4	5	6	7	1

Since it's rotated we only need to search right half

nums[mid-1] > nums[mid] only 7 1 (mid 2)

nums[mid] > nums[mid+1] only 6 7 1

else update mid

if nums[mid] > nums[left].

↳ we only need to look at right side.

left = mid+1

else

right = mid-1

Then,

6	7	1
l	mid	r

nums[mid] > nums[mid+1]

return nums[mid+1]