Find Minimum in Rotated Sorted Array

Difficulty	Medium
Category	Binary Search
Question	https://leetcode.com/problems/find-minimum-in-rotated-sorted-array/
Solution	https://youtu.be/nIVW4P8b1VA
Status	Done

Question

> Suppose an array of length `n` sorted in ascending order is **rotated** between `1` and `n` times. For example, the array `nums = [0,1,2,4,5,6,7]` might become:
>
> - `[4,5,6,7,0,1,2]` if it was rotated `4` times.
> - `[0,1,2,4,5,6,7]` if it was rotated `7` times.
>
> Notice that **rotating** an array `[a[0], a[1], a[2], ..., a[n-1]]` 1 time results in the array `[a[n-1], a[0], a[1], a[2], ..., a[n-2]]`.
>
> Given the sorted rotated array `nums` of **unique** elements, return *the minimum element of this array*.

Example

Example 1:

```
Input: nums = [3,4,5,1,2]
Output: 1
```

```
Explanation: The original array was [1,2,3,4,5] rotated 3 times.
```

Example 2:

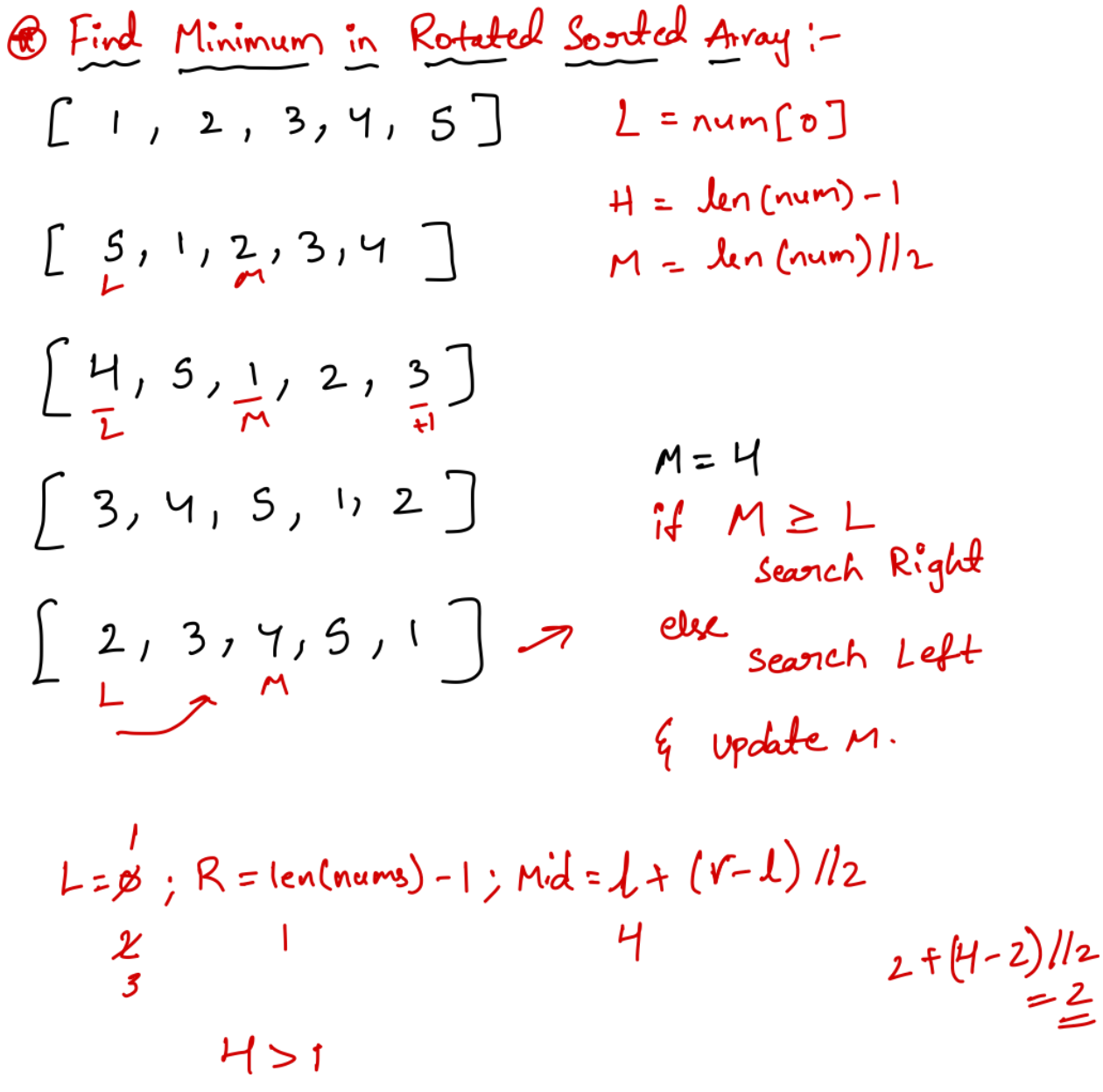
```
Input: nums = [4,5,6,7,0,1,2]
Output: 0
Explanation: The original array was [0,1,2,4,5,6,7] and it was rotated 4 times.
```

Example 3:

```
Input: nums = [11,13,15,17]
Output: 11
Explanation: The original array was [11,13,15,17] and it was rotated 4 times.
```

Idea

Find the pivot where nums[i] is smaller than nums[i-1]



Solution

```
class Solution:
    def findMin(self, nums: List[int]) -> int:
    # Initialize two pointers, left and right, to the start and end of the array.
        left, right = 0, len(nums) - 1

        # Perform binary search while left pointer is less than right pointer.
        while left < right:
            # Calculate the middle index.
            mid = left + (right - left) // 2
```

```
            # Check if the middle element is greater than the right element.
            if nums[mid] > nums[right]:
                # If true, the minimum element must be on the right side of mid.
                left = mid + 1
            else:
                # If false, the minimum element is on the left side of mid or is mid itself.
                right = mid

        # At the end of the loop, left and right will be pointing to the minimum element.
        return nums[left]
```

Explanation