

Min in Sort rotated Array

Imagine you're working on an update system for a fleet of autonomous vehicles at Uber. The system stores updates in a sorted array for efficiency, but due to unexpected reboots, the array sometimes gets **rotated** at a pivot point. Your task is to quickly identify the **first version** (the minimum value) after a rotation.

Problem Statement:

Given an array that was initially sorted in ascending order but then rotated at an unknown pivot, write an efficient algorithm to find the **minimum element** in the array.

- The array contains **no duplicate elements**.
- Your solution must run in **$O(\log N)$** time.

Input Format:

- A single integer N denoting the number of elements in the array.
- A line containing N space-separated integers, representing the rotated sorted array.

Output Format:

- Print a single integer — the minimum element in the rotated array.

Constraints:

- $1 \leq N \leq 10^5$
- $-10^9 \leq A[i] \leq 10^9$
- No duplicate elements.
- The array is a rotated version of a sorted array.

Example:

Input:

```
5
5 7 10 3 4
```

Output:

```
3
```

Explanation:

The original array before rotation might have been [3, 4, 5, 7, 10]. After rotation, it becomes [5, 7, 10, 3, 4]. The minimum element is 3.

Approach (Binary Search):

- Initialize `low = 0, high = N - 1`
- While `low < high`:
 - Find `mid = (low + high) / 2`
 - If `arr[mid] > arr[high]`, the minimum is in the **right half** $\rightarrow low = mid + 1$
 - Else, the minimum is in the **left half (including mid)** $\rightarrow high = mid$
- Return `arr[low]`

 **Time Complexity:** $O(\log N)$

 **Space Complexity:** $O(1)$

Additional Test Cases:

Input	Output
6\n6 7 8 1 2 3	1
4\n2 3 4 1	1
5\n1 2 3 4 5	1
5\n4 5 1 2 3	1
1\n99	99

Practice Links:

- LeetCode: [Find Minimum in Rotated Sorted Array](#)
- GFG: [Minimum Element in a Rotated Sorted Array](#)

Video Solution:

- [YouTube - Intuition & Binary Search](#)
- [Binary Search Rotated Array Explanation](#)