Problem of the Week – Fixed Point in a Sorted Array

Company: Apple
Difficulty: Medium
Topic: Binary Search, Arrays

📌 Scenario

In algorithmic problem solving, a **fixed point** in an array is an index `i` such that:

```
arr[i] == i
```

Apple engineers use this property in certain optimization checks.
You are given a **sorted array of distinct integers**, and you need to find if such a fixed point exists.

📌 Problem Statement

Given a **sorted array of distinct integers**, return **any fixed point** if it exists, otherwise return **False**.

🔷 Input Format

- First line: integer `N` – size of the array.
- Second line: `N` space-separated integers (sorted, distinct).

🔷 Output Format

- Print the fixed point if it exists, otherwise print `False`.

🔷 Constraints

- `1 ≤ N ≤ 10^5`
- Array elements can be negative, zero, or positive.
- Elements are distinct and sorted in increasing order.

🔷 Examples

Sample Input 0

```
4
-6 0 2 40
```

Sample Output 0

```
2
```

Explanation 0
At index `2`, `arr[2] = 2`. Hence, the fixed point is `2`.

Sample Input 1

```
4
1 5 7 8
```

Sample Output 1

```
False
```

Explanation 1
No index `i` satisfies `arr[i] == i`.

🔑 Approaches

1. **Brute Force (O(N))**
   - Iterate through the array and check if `arr[i] == i`.
   - Return the first such value if found, else `False`.
2. **Efficient Binary Search (O(log N))**
   - Since the array is sorted and distinct:
     - If `arr[mid] == mid`, return `mid`.
     - If `arr[mid] > mid`, search **left half**.
     - If `arr[mid] < mid`, search **right half**.

🔗 Practice Links

- GeeksforGeeks – Fixed Point in an array
- LeetCode Discussions – Fixed Point Problem