Longest Increasing Subsequence (LIS)

Title – Longest Increasing Subsequence

Company: Microsoft
Difficulty: Medium
Topic: Dynamic Programming

📌 Problem Statement

Given an array of numbers, find the length of the **Longest Increasing Subsequence (LIS)**. The subsequence does **not** need to be contiguous, but the order must be maintained.

🔷 Example

Input:

```
[0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15]
```

Output:

```
6
```

Explanation:
The LIS is `[0, 2, 6, 9, 11, 15]` of length 6.

🔑 Approaches

1. **Recursive + Memoization (Top-Down DP):**
   - Try including or excluding each element.
   - Memoize results to avoid recomputation.
   - Time Complexity: `O(n^2)`
2. **Bottom-Up DP (Classic DP):**
   - Use an array `dp[i]` = LIS ending at index `i`.
   - Transition: `dp[i] = 1 + max(dp[j])` for all `j < i` where `arr[j] < arr[i]`.
   - Time Complexity: `O(n^2)`
3. **Optimized Approach with Binary Search (Patience Sorting Method):**
   - Maintain a temp array.
   - For each number:
     - If greater than the largest element, append it.
     - Else, replace the smallest element ≥ current number.

    - Time Complexity: `O(n log n)`

🔗 Practice Links

- LeetCode – Longest Increasing Subsequence
- GeeksforGeeks – Longest Increasing Subsequence