

UNITED INTERNATIONAL UNIVERSITY
DSA 2 Lab Section C
Coding Test 3 (Dynamic Programming)
Total: 10 Time: 25 minutes

Problem Statement:

Given an array of integers '*arr*', find the length of the **longest increasing subsequence (LIS)** in the array. A subsequence of an array is a sequence of elements that can be obtained by removing some or no elements from the array, but maintaining the relative order of the remaining elements. An increasing subsequence is a subsequence in which the elements are in increasing order.

For example, given the array '*[10, 9, 2, 5, 3, 7, 101, 18]*', the longest increasing subsequence is '*[2, 3, 7, 101]*', which has a length of 4.

Write a function '*longest_increasing_subsequence(arr: List[int]) -> int*' that takes in an array of integers and returns an integer representing the length of the longest increasing subsequence in the array.

Constraints:

- The length of the input array is between 1 and 2500.
- The elements of the input array are between -10^4 and 10^4 .

Hint:

You can solve this problem using dynamic programming by defining a 1D array '*dp*' of the same length as the input array, where '*dp[i]*' represents the length of the longest increasing subsequence that ends at index '*i*'. You can then iterate through the input array, and for each index '*i*', compute '*dp[i]*' as the maximum of '*dp[j] + 1*' for all '*j*' less than '*i*' and such that '*arr[j] < arr[i]*'. The final answer is the maximum value in the '*dp*' array.