**Longest Increasing Subsequence –DHAIRYASHIL PATIL**

**Problem Statement:** **As a developer, write a Java code to find the longest increasing subsequence from a list of random numbers.**

**Algorithm:**

1. Initialize an[array](https://www.tutorialcup.com/java/arrays-in-java.htm) a[ ] of [integer](https://www.tutorialcup.com/java/data-types-in-java.htm) type of size n.
2. Create a function to find number of the [longest increasing sub-sequences](https://www.tutorialcup.com/interview/array/longest-increasing-subsequence-2.htm) which accept an array of integer type and it’s size as it’s parameters.
3. Create two arrays of integer type len and cnt of size n and initialize every element of both the arrays as 1. Also, initialize an integer variable lis = 1.
4. Traverse from 1 till n-1 using i and create an inner loop from 0 to i-1.
5. Check if the element in array a[ ] at current index of outer loop is greater than the element in array a[ ] at the current index of the inner loop, check if the value + 1 in array len[ ] at current index of inner loop is greater than the element in array len[ ] at current index of outer loop, update the value in array len[ ] at current index of outer loop as the value + 1 in array len[ ] at current index of inner loop and the value in array cnt[ ] at current index of outer loop as the value in array cnt[ ] at current index of inner loop.
6. Else if the value + 1 in array if len[ ] at current index of inner loop is equal to the element in array len[ ] at current index of outer loop, update the value in array cnt[ ] at current index of outer loop as the sum of the value in array cnt[ ] at current index of inner loop and the outer loop itself.
7. Store the maximum of lis and len[i] in lis.
8. Initialize a variable ans as 0.
9. Traverse again from 0 to n-1 and check if len[i] is equal to lis add the value at the current index of cnt in ans.
10. Return ans.