**-------------------------------------------------------------**

**NAME:--** RAVI KASHYAP

**GitHubLink:**

**Background of the problem statement:---**

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DESCRIPTION**

In this project, A subsequence is derived from an arrayList by deleting a few of its elements and not changing the order of remaining elements. We have given an arrayList AL with N elements. Increasing subsequence is a subsequence with its elements in increasing order. We need to find the lengthof the longest increasing subsequence that can be derived from the given arrayList.

**Time complexity:** O(2^n), where n is the size of array.

**Space** **complexity:** O(n) for stack space in recursion.

**An Efficient Algorithm for Find the Longest Increasing Subsequence:--**

* Iterative function to find the longest increasing subsequence of a given array**public** **int** lengthOfLIS(ArrayList arrayList)
* . Base case

**if** (arrayList.size() == 0) {

System.***out***.println("Longest Increasing Subsequence is :" + 0);

**return** 0;

* LIS[i] stores the longest increasing subsequence of subarray

         `arr[0…i]` that ends with `arr[i]`

ArrayList<Integer> newArraylist = **new** ArrayList<Integer>();

**for** (**int** x = 0; x < size; x++) {

newArraylist.add(0);

* Start from the second array element.

**for** (**int** i = 1; i < newsize; i++) {

**int** newValueMax = 0;

* Do for each element in subarray `arr[0…i-1]

**for** (**int** j = 0; j < i; j++)

* find the longest increasing subsequence that ends with `arr[j]`

where`arr[j]` is less than the current element `arr[i]`

**if** ((**int**) arrayList.get(i) > (**int**) arrayList.get(j)) {

newValueMax = Math.*max*(newValueMax, newArraylist.get(j));

}

* include `arr[i]` in `LIS[i]`

newArraylist.add(i, newValueMax + 1);

maximum = Math.*max*(maximum, newArraylist.get(i));

* `j` will store an index of **the Longest Increasing Subsequence(LIS)**

**int** listSize = sc.nextInt();

**for** (**int** i = 0; i < listSize; i++) {

System.***out***.println("Enter value for index #" + i + " :");

**int** value = sc.nextInt();

myArray.add(value);

* Print Longest increasing Subsequence

System.***out***.println("Your list of numbers is: " + myArray);

LongestArraySub lis = **new** LongestArraySub();­­­­­­­­

lis.lengthOfLIS(myArray);

**OUTPUT:--**

Please enter the size of you list:

5

Enter value for index #0 :

5

Enter value for index #1 :

4­­­­­­

Enter value for index #2 :

3

Enter value for index #3 :

2

Enter value for index #4 :

9

Your list of numbers is: [5, 4, 3, 2, 9]

Longest Increasing Subsequence is :2