

30.03.2022

# LONGEST INCREASING SUBSEQUENT PROJECT

Prepared by: PRIYANKA DAS

# Project objective:

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

## We must use the following:

Eclipse/IntelliJ: An IDE to code the application

Java: A programming language

Git: To connect and push files from the local system to GitHub

GitHub: To store the application code and track its versions

The code should work properly for  $n$  numbers, where  $n < 100$ .

Following requirements should be met:

The versions of the code should be tracked on GitHub repositories

The code should be able to search the required string from the array of strings

# About Project

The Longest Increasing Subsequence (LIS) problem is to find the length of the longest subsequence of a given sequence such that all elements of the subsequence are sorted in increasing order.

For example, the length of LIS for {10, 22, 9, 33, 21, 50, 41, 60, 80} is 6 and LIS is {10, 22, 33, 50, 60, 80}.

Example:

Input: arr[] = {5,4,1,2,3}

Output: Length of LIS = 3

Explanation: The longest increasing subsequence is 1,2,3

Input: arr[] = {7,5}

Output: Length of LIS = 1

Explanation: The longest increasing subsequences are {5} or {7}.

# About Project Code

At first create a class, then performing the following steps :-

- > filling the array with random integers.
- > check for null or empty array
- > a set is used because a subsequence should not contain duplicate elements  
for example: [1,2,2,2,3] should be saved as [1,2,3]

My interpretation of the problem is that only a lesser integer should break a sequence.

- > iterate over the array while keeping track of the value of the previous element.
- > if the current number is greater than the last, add them to the current set.
- > The largest length so far will either be the current length or the last length.
- > In this case the current set isn't increasing any further,  
so it is added to the total list of subsequences and the current set is cleared.

# About Project

## Code(contd)

--> Save the current subsequence when we have reached the end of the array.

--> The largest subsequence is picked from the total list of sequences.

In a situation where multiple subsequences are the largest length, the first subsequence will be chosen.

for example:

- input: [1,2,3,0,5,2,3,1,5,6]

- output: [1,2,3], length 3

--> If all of the numbers in the array are decreasing, then there is no subsequence of increasing numbers.

My interpretation is that 1 should still be returned for the largest length of the sequence.