**Source code for Longest Increasing Subsequence**

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
  
public class LongestIncreasingSubsequence {  
  
 public static List<Integer> findLIS(int[] nums) {  
 if (nums == null || nums.length == 0) {  
 return new ArrayList<>();  
 }  
  
 int[] dp = new int[nums.length];  
 Arrays.*fill*(dp, 1);  
  
 for (int i = 1; i < nums.length; i++) {  
 for (int j = 0; j < i; j++) {  
 if (nums[i] > nums[j] && dp[i] < dp[j] + 1) {  
 dp[i] = dp[j] + 1;  
 }  
 }  
 }  
  
 int maxLength = 0;  
 int endIndex = 0;  
  
 for (int i = 0; i < dp.length; i++) {  
 if (dp[i] > maxLength) {  
 maxLength = dp[i];  
 endIndex = i;  
 }  
 }  
  
 List<Integer> lis = new ArrayList<>();  
 lis.add(nums[endIndex]);  
  
 for (int i = endIndex - 1; i >= 0; i--) {  
 if (nums[i] < nums[endIndex] && dp[i] == dp[endIndex] - 1) {  
 lis.add(nums[i]);  
 endIndex = i;  
 }  
 }  
  
 return lis;  
 }  
  
 public static void main(String[] args) {  
 int[] nums = {100, 22, 9, 33, 258, 50, 41, 60, 80 , 1 , 10 , 21};  
 List<Integer> lis = *findLIS*(nums);  
  
 System.*out*.println("Length of Longest Increasing Subsequence: " + lis.size());  
 System.*out*.println("Longest Increasing Subsequence: " + lis);  
 }  
}