```
import java.util.*;
public class LIS {
    public static List<Integer> longestsub(int[] nums) {
        int n = nums.length;
        int[] l1 = new int[n];
        int[] l2 = new int[n];
        for (int i = 0; i < n; i++) {
            l1[i] = 1;
            l2[i] = -1;
            for (int j = 0; j < i; j++) {
                if (nums[i] > nums[j] && l1[j] + 1 > l1[i]) {
                    l1[i] = l1[j] + 1;
                    l2[i] = j;
                }
            }
        }
        int maxLength = 0;
        int maxIndex = -1;
        for (int i = 0; i < n; i++) {
            if (l1[i] > maxLength) {
                maxLength = l1[i];
                maxIndex = i;
            }
        }
        List<Integer> l3 = new ArrayList<>();
        while (maxIndex != -1) {
            l3.add(0, nums[maxIndex]);
            maxIndex = l2[maxIndex];
        }
        return l3;
    }
    public static void main(String[] args) {
        int[] nums = {11, 23, 90, 33, 2, 50, 41, 60, 80};
        List<Integer> l3 = longestsub(nums);
        System.out.println("Longest Increasing Subsequence: " + l3);
    }
}
```