Longest Increasing Subsequence

Source code

```java
package practiceProject2;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class LongestIncreasingSubsequence {
    public static void main(String[] args) {
        int[] numbers = {10, 22, 9, 33, 21, 50, 41, 60, 80};
        List<Integer> lis = findLongestIncreasingSubsequence(numbers);

        System.out.println("Longest Increasing Subsequence: " + lis);
    }

    public static List<Integer> findLongestIncreasingSubsequence(int[] numbers) {
        int n = numbers.length;
        int[] dp = new int[n];
        int[] prev = new int[n];

        Arrays.fill(prev, -1);

        dp[0] = 1;

        for (int i = 1; i < n; i++) {
            dp[i] = 1;

            for (int j = 0; j < i; j++) {
                if (numbers[i] > numbers[j] && dp[i] < dp[j] + 1) {
                    dp[i] = dp[j] + 1;
                    prev[i] = j;
                }
            }
        }

        int maxLength = dp[0];
        int maxIndex = 0;

        for (int i = 1; i < n; i++) {
            if (dp[i] > maxLength) {
                maxLength = dp[i];
                maxIndex = i;
            }
        }

        List<Integer> lis = new ArrayList<>();

        while (maxIndex >= 0) {
            lis.add(numbers[maxIndex]);
            maxIndex = prev[maxIndex];
        }

        lis.sort(null); // Sort the lis in ascending order
        return lis;
    }
}
```