**Sourcecode:**

**package** practiseproj;

**import** java.util.Arrays;

**public** **class** Lincsubseq {

**public** **static** **int**[]

findLincsubseq(**int**[] nums) {

**int** n = nums.length;

**int**[] dp = **new** **int**[n];

**int**[] prevIndex = **new** **int**[n];

Arrays.*fill*(dp, 1);

Arrays.*fill*(prevIndex, -1);

**int** maxLength = 1;

**int** lastIndex = 0;

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

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

**if** (nums[i] > nums[j] && dp[i] < dp[j] + 1) {

dp[i] = dp[j] + 1;

prevIndex[i] = j;

}

}

**if** (dp[i] > maxLength) {

maxLength = dp[i];

lastIndex = i;

}

}

**int**[] longestSubsequence = **new** **int**[maxLength];

**int** currentIndex = maxLength - 1;

System.***out***.println("length os lis is:"+maxLength);

**while** (lastIndex >= 0) {

longestSubsequence[currentIndex] = nums[lastIndex];

currentIndex--;

lastIndex = prevIndex[lastIndex];

}

**return** longestSubsequence;

}

**public** **static** **void** main(String[] args) {

**int**[] nums = {5,10,8,345,32,53,2322,64,94,100,80,150,164,84,96,200};

**int**[] longestSubsequence =*findLincsubseq*(nums);

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

**for** (**int** num : longestSubsequence) {

System.***out***.print(num + " ");

}

}

}