





（4）思路：设数组为a[0:n], 用数组 b[0:n]纪录以 a[i] (0<= i< n)，为结尾元素的最长递增子序列的长度。则对于序列中第一个数a[0]，显然有b[0]=1，之后可以从左向右按顺序计算a[i]（i=1，2，3，…… ，n）作为结尾元素时，最长递增子序列的长度b[i]。当计算到 a[i]的最长递增子序列时，应该选择比a[i]小的数中以该数结尾的递增子序列长度最长的那个数，作为子序列的倒数第二个数，即b[i]=max{b[j]+1}（0≤ j < i且 a[j] < a[i]）;如果序列a[0:i-1]中所有的数均不小于a[i]，则令b[i]=1，即以a[i]为结尾元素时的最长递增子序列只包含a[i]本身。

伪代码：

LongestSubstr(a[0:n]){

for( i = 1 to n){

max = 0;

for( j = 0 to i){

if(a[j] < a[i]){

if(max < b[j]){

max = b[j];

}

}

}

b[i] = max + 1;

}

最后找到b数组中的最大值（下标记为k）即为最长递增子序列的长度。从a[k]开始，从右向左依次遍历数组a[n]。对于某个下标l（0 ≤ l < k）,如果l满足a[l] < a [k]且b[l] = b[k] - 1，则a[l]可以作为最长递增子序列的倒数第二个数。以此类推，可以重复直至得到最长递增子序列。