

## 1 程序设计思路

问题：有长度为  $n$  的  $A = \{a_1, a_2, \dots, a_n\}$  的序列，求其最长的严格递增子序列。

假定以  $a_j (1 \leq j \leq n)$  结尾的最长递增子序列的长度为  $L(j)$ ，则有  $L(j) = \max(L(i)) + 1, (i < j) \&\& (a_i < a_j)$ 。因此只要遍历序列的所有元素，找到以其结尾的最长递增子序列长度，比较得到最长的长度，得到对应的 LIS 即可。

根据上述思路实现的伪代码如下：

```
LIS(a):
    // 得到每个元素为结尾的LIS的长度，存在maxlen数组中
    int maxlen[a.length];
    for i = 0 to (a.length - 1)
        maxlen[i] = 1;
    for i = 1 to (a.length - 1) {
        for j = 0 to (i - 1) {
            if ((a[i] > a[j]) && (maxlen[i] < maxlen[j]+1)){
                maxlen[i] = maxlen[j] + 1;
            }
        }
    }
    // 找到最长的LIS对应的元素下标
    index = 0;
    for i = 1 to (a.length - 1)
    {
        if (maxlen[i] > maxlen[index])
            index = i;
    }
    // 得到最长的LIS，存放在lis数组中
    int lis[maxlen[index]];
    j = 0;
    for i = 0 to (index-1) {
        if (maxlen[i] == (j+1)) {
            lis[j] = a[i];
            ++j;
        }
    }
}
```

对上述伪代码进行复杂度分析，假设序列长度为  $n$ ，则第一个 for 循环的复杂度为  $O(n^2)$ ，之后的 for 循环的复杂度均为  $O(n)$ 。故程序的复杂度为  $O(n^2)$ 。

## 2 直观的例子

假设序列为  $A = \{5, 6, 7, 2, 0, 9, 3\}$ 。则以每个元素为结尾的 LIS 的长度为  $maxlen = \{1, 2, 3, 1, 1, 4, 2\}$ 。其中最大的数字是 4，对应的  $index = 5$  (假设序列标号从 0 开始)，那么就可以找到最长递增序列为  $lis = \{5, 6, 7, 9\}$ 。