**算法设计与分析网络学习第七次课程学习指南**

**时间**：2020年3月16日(星期二)上午10:00（3-4节）

**课堂安排**： 10:00-11:00观看视频，做练习题和（<https://www.icourse163.org/course/HIT-356006>）

**（**<https://mooc.study.163.com/learn/1000005000?tid=1000005001#/learn/content>**）**

11:00-11:45 强调知识点、答疑、讨论

(腾讯会议号码：686 206 168)

**学习内容**：动态规划的典型问题

**本次课程习题助教**：陈剑

**1. 授课视频**

算法设计与分析(基础篇) 4.3

算法设计与分析(进阶篇) 2.4, 2.5

**2. 阅读**

算法导论(第三版) 第15.4节， 35.5节(选学)

**3. 练习题**

**3.1** 给定一个浮点数序列（可能有正数、0和负数），设计一个动态规划算法，求出一个最大的连续子序列中浮点数乘积。

（1）简明的写出问题的递推方程；

（2）描述算法伪代码；

（3）分析算法的时间复杂度。

**3.2** 已知一个矩形区域被划分为N\*M个小矩形格子，在格子(i,j)中有A[i][j]个苹果。现在从左上角的格子(1,1)出发，要求每次只能向右走一步或向下走一步，最后到达(N,M)，每经过一个格子就把其中的苹果全部拿走。设计一个动态规划算法，找出能拿到最多苹果数的路线。

（1）简明的写出问题的递推方程；

（2）描述算法伪代码；

（3）分析算法的时间复杂度。

（4）将原问题变为“从起点出发，用相同的方法走两次，但是两条路线中除了起点和终点外不经过任何同一节点”，简明写出动态规划解决这个新问题的思路和递推方程。

**3.3** 考虑三个字符串X,Y,Z 的最长公共子序列LCS(X,Y,Z)。

(1)寻找反例X,Y,Z 使得LCS(X,Y,Z)≠LCS(X, LCS(Y,Z));

(2)设计动态规划算法计算X,Y,Z 的最长公共子序列,分析算法的时间复杂度。

**3.4** 令*I1, …, In*是*n*个区间，其中任一区间*Ii=(ai,bi)*，假设这些区间按照*bi*从小到大排序，每一个区间有一个权重*vi*,考虑如下两个问题：

区间安排问题 P1: 找到最大数量互不相交的区间，例如，对于四个区间*I1 = (1,2)*; *I2 = (2,3)*; *I3 = (1,4)*; *I4 = (4,5)*，一个解是*{I1, I2, I4}*

加权任务安排问题P2: 找一个互不相交区间的集合，使得这些区间的权重之和最大，例如*I1 = (1,2)*, *v1=0.9*; *I2 = (2,3), v2=0.5; I3 = (1,4), v3=4; I4 = (4,5), v4=2*，解是*{I3, I4}*。

(1) 给出解决问题P1的线性时间算法。

(2) 给出解决问题P2的动态规划算法，要求写出递归方程和伪代码，并分析算法时间复杂性。

**3.5** 编辑距离问题

**(a)** 设计动态规划算法计算字符串s1和s2的扩展编辑距离。字符串s1和s2的扩展编辑距离是s1变换到s2的最少操作个数，其中操作包括1)插入一个字符；2)删除一个字符；3)修改一个字符；4)交换两个相邻的字符。要求写出解的递推方程、算法的伪代码并分析算法的时间复杂度。

**(b)** 考虑字符串变换操作，增加一个字符，删除一个字符以及修改一个字符，设增加字符操作的代价为i, 删除字符操作代价为d, 修改字符的代价为m，给定两个字符串S1和S2，设计一个动态规划算法，求得从S1变换到S2代价最小的变换序列，要求写出递推方程，程序伪代码并分析算法复杂性。

**(c)** 考虑编辑距离的一种变形，其允许在字符串后无代价地插入无限多个字符，该编辑距离描述为:

ed'(A, B)=min{ed(A, C)|C是B的前缀}, 其中函数ed()是普通的编辑距离函数。根据要求设计算法，要求算法的时间复杂性都是O(|A||B|)

(1) 设计算法，对于给定的字符串A和B，计算ed'(A, B)；

(2) 设计算法，对于给定的字符串A,B和整数k，判定是否B存在某个后缀B’，满足ed’(A, B’)≤k。

**3.6** T1和T2是两棵有序树，其中每个结点都有一个标签，考虑树上的三种操作，删除一个子树、插入一个子树和更改一个结点的标签，请设计一个算法，求得从T1变化到T2所需要的最少操作数，要求写出递推方程，程序伪代码并分析时间复杂性。

**4. 思考题**

动态规划算法的子结构都有哪些种类？