**算法设计与分析网络学习第八次课程学习指南**

**时间**：2020年3月19日(星期四)上午10:00（3-4节）

**课堂安排**： 10:00-11:00观看视频，做练习题和思考题

**（**<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：311115337；密码：090405)

**学习内容**：动态规划的子结构

**本次课程习题助教**：黄河凯

**1. 授课视频**

算法设计与分析(进阶篇) 2.1, 2.3

**2. 阅读**

算法导论(第三版)

**3. 练习题**

**3.1** 设计动态规划算法输出数组A[0:n]中的最长单调递增子序列。

**3.2** 给定n种物品和一背包，物品i的重量是wi,其价值是vi, 背包的容量是c, 设计一个完全多项式时间近似算法，求装入背包的物品集合，使得装入背包中物品的总价值最大。要求设计出算法并对其近似比进行分析，说明为何此算法是完全多项式时间近似模式的算法。

**3.3** 考虑特殊的0-1背包问题：有n个物品，每个物品i价值和重量都是wi，背包能容纳物品的最大重量是C, 选择背包能容纳的物品集合，使得这些物品价值之和最大。回答下列问题:

1. 若物品的重量(价值)分别是1, 2, …,2n-1, 证明该0-1背包问题可以用贪心法求解并写出该贪心算法的伪代码。
2. 请写出一个物品重量(价值)序列，使得上述贪心法无法得到最优解。

**3.4** 正整数n 可以拆分成若干个正整数之和，考虑拆分方案的个数。

(1) 令g(i,j)表示拆分整数i时最大加数不超过j的方案个数， 证明：g(i,j)=g(i,j-1)+g(i-j,j)。

(2)根据(1)设计动态规划算法计算整数n的拆分方案个数，要求算法的时间复杂度为O(n2)

**3.5.** 设计时间复杂度为*O*(*n*2*k*)的动态规划算法，找到最优聚类。要求写出伪代码、递归方程并分析算法的时间复杂度。

例如，考虑将4个元素的集合{1,5,8,10}聚为两个类，有三种可能:

1. S1={1}, S2={5,8,10}，总代价是02+52=25
2. S1={1,5}, S2={8,10}，总代价是42+22=20
3. S1={1,5,8}, S2={10}，总代价是72+02=49

所以，算法的解是最优解S1={1,5}, S2={8,10}。

**3.6** 我们考虑将数轴上的*n*个点聚成*k*类的问题。

输入：*n*个从小到大的不同实数*x*1, *x*2, …, *xn*表示*n*个不同点，一个参数*k*≤*n*.

任务：将*n*个点划分成*k*个不相交的非空集合*S*1, …., *Sk*满足={*x*1, *x*2, …, *xn*}，*Si*中所有点在*Si*+1中所有点左边，1≤*i*<*k*，也就是说对于任意*x*∈*Si*, *z*∈*Si*+1, *y*<*z*.

目标：最小化,其中cost(*Si*)=(max(*Si*)-min(*Si*))2. max(*Si*)是*Si*中的最小元素，min(*Si*)是*Si*中的最大元素。

例如，如果*Si*={xj}，cost(*Si*)=0，如果*Si*={*xj*, *xj*+1, …, *xj+t*}, *xj*,<*xj*+1< …< *xj+t*，那么cost(*Si*)=(*xj+t*-*xj*)2.

设计时间复杂度为*O*(*n*2*k*)的动态规划算法，找到最优聚类。要求写出伪代码、递归方程并分析算法的时间复杂度。

例如，考虑将4个元素的集合{1,5,8,10}聚为两个类，有三种可能:

1. S1={1}, S2={5,8,10}，总代价是02+52=25
2. S1={1,5}, S2={8,10}，总代价是42+22=20
3. S1={1,5,8}, S2={10}，总代价是72+02=49

所以，算法的解是最优解S1={1,5}, S2={8,10}。

**4. 思考题**

动态规划算法的子结构能否进行压缩？