第四章　动态规划习题

1. 给定一个整数序列 $a_1,\ldots,a_n$。相邻两个整数可以合并，合并两个整数的代价是这两个整数之和。通过不断合并最终可以将整个序列合并成一个整数，整个过程的总代价是每次合并操作代价之和。试设计一个动态规划算法给出 $a_1,\ldots,a_n$ 的一个合并方案使得该方案的总代价最大。

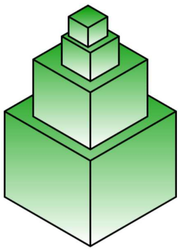
2. 输入表达式 $a_1O_1\ a_2\ O_2\ldots\ldots O_{n-1}\ a_n$，其中 a_i 是整数$(1\le i\le n)$,$O_j\in\{+,-,\times\}(1\le i\le n-1)$。

   (1)试设计一个动态规划算法，输出一个带括号的表达式（不改变操作数和操作符的次序），使得表达式的值达到最大，分析算法的时间复杂性。

   (2)令 $O_j\in\{+,-,\times,\div\}$,重新完成(1)规定各项任务。

3. 设平面上有一个 $m\times n$ 的网格，将左下角的网格点标记为（0,0）而右上角的网格点标记为（m,n）。某人想从（0,0）出发沿网格线行进到达（m,n），但是在网格点（i,j）处他只能向上行进或者向右行进，向上行进的代价为 a_{ij}（a_{mj} ＝＋∞），向右行进的代价是 b_{ij}（b_{in} ＝＋∞）。试设计一个动态规划算法，在这个网格中为该旅行者寻找一条代价最小的旅行路线。

4. 给定 n 个长方体盒子，第 i 个盒子的长、宽、高分别为 l_i, w_i, h_i。盒子允许旋转。如果上方盒子的底(长、宽均)小于等于下方盒子的底(长、宽)，则两个盒子可以堆叠起来。试设计一个动态规划算法，将所有盒子堆叠得尽可能高，你需要：(1)分析问题的优化子结构；(2)分析问题的子问题重叠性；（3）给出代价的递归方程；（4）描述算法；（5）分析算法的时间复杂度和空间复杂度。



思考题(共参考，无需提交)

4.1 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 $O(n^2)$。

4.2 设 $R(X)$表示将整数 X 的各个数位取逆序后得到的整数，如 $R(123)=321$, $R(120)=21$。现输入正整数 N，试设计一个动态规划算法计算方程 $R(X)+X=N$ 的解的个数，分析算法的时间复杂度。

4.3 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。

4.4 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。

4.5 输入数组 $A[0:n]$和正实数 d,试设计一个动态规划算法输出 $A[0:n]$的一个最长子序列，使得子序列中相继元素之差的绝对值不超过 d。分析算法的时间复杂度。

4.6 给定一个 $n \times n$ 的矩阵 A，矩阵中的元素只取 0 或者 1。设计一个动态规划算法，求解得到 A 中元素全是 1 的子方阵使其阶数达到最大值。

4.7 集合划分问题描述如下：

输入：正整数集合 $S=\{a_1, a_2, a_3, \ldots, a_n\}$；

输出：是否存在 $A \subseteq S$ 使得 $\sum_{a_i \in A} a_i = \sum_{a_i \in S-A} a_i$；

试设计一个动态规划算法，求解集合划分问题。

4.8 输入平面上 n 个点，点与点之间的距离定义为欧几里得距离。试设计一个动态规划算法输出一条先从左到右再从右到左的一条最短路径，使得每个输入点恰好被访问一次。

4.9 输入凸 n 边形 $p_1, p_2, \ldots p_n$,其中顶点按凸多边形边界的逆时针序给出，多边形中不相邻顶点间的连线称为弦。试设计一个动态规划算法，将凸边形 $p_1, p_2, \ldots p_n$ 剖分成一些无公共区域的三角形，使得所有三角形的周长之和最小。

4.10 输入一棵加权树 T,其中每条边的权值均是实数，试设计一个动态规划算法输出权值最大的子树。