第十五章动态规划习题答案

1．写出以下背包问题实例的求解过程（递归、元组法）

n=5, P=[6,3,5,4,6],w=[2,2,6,5,4]，c=10

解: $f(5,y)=\begin{cases}6, & y\ge 4\\ 0, & y<4\end{cases}$ P[5]=[(0,0),(4,6)]

$$f(4,y)=\begin{cases}\max(f(5,y),f(5,y-5)+4), & y\ge w_4\\ f(5,y), & y<w_4\end{cases}$$

$$=\begin{cases}6, & 4\le y<5\\ 0, & y<4\\ 6, & 5\le y<9\\ 10, & y\ge 9\end{cases}$$ P[4]=[(0,0),(4,6),(9,10)]

$$=\begin{cases}0, & y<4\\ 6, & 4\le y<9\\ 10, & y\ge 9\end{cases}$$

$$f(3,y)=\begin{cases}\max(f(4,y),f(4,y-6)+5), & y\ge w_3\\ f(4,y), & y<w_3\end{cases}$$

$$=\begin{cases}0, & y<4\\ 6, & 4\le y<6\\ 6, & 6\le y<9\\ 10, & 9\le y<10\\ 11, & y\ge 10\end{cases}$$ p[3]=[(0,0),(4,6),(9,10),(10,11)]

$$=\begin{cases}0, & y<4\\ 6, & 4\le y<9\\ 10, & 9\le y<10\\ 11, & y\ge 10\end{cases}$$

$$f(2,y)=\begin{cases}\max(f,(3,y),f(3,y-2)+3,y\ge w_2\\ f(3,y), & y<w_2\end{cases}$$

$$=\begin{cases}0, & y<2\\ 3, & 2\le y<4\\ 6, & 4\le y<6\\ 9, & 6\le y<9\\ 10, & 9\le y<10\\ 11, & y=10\end{cases}$$

p[2]=[(0,0),(2,3),(4,6),(6,9),(9,10),(10,11)]
使用元组法求解的过程：
1）P[5]=[(0,0),(4,6)]
2）Q＝[(5,4),(9,10)],因为(4,6)支配(5,4),所以消去(5,4)
P[4]=[(0,0),(4,6),(9,10)]
3）Q=[(6,5),(10,11)],因为(4,6)支配(6,5)，所以消去(6,5)
P[3]=[(0,0),(4,6),(9,10),(10,11)]
4）Q=[(2,3),(6,9)],
P[2]=[(0,0),(2,3),(4,6),(6,9),(9,10),(10,11)]
5）f(1,10)=max{f(2,10),f(2,10-2)+6}=max{f(2,10),f(2,8)+6}=max{11,9+6}=15

回溯求优化解：
c=10, f(1,10)=15, f(2,10)=11,所以 x1=1,c=10-2=8
f(2,8)=9,f(3,8)=6,所以 x2=1,c=8-2=6
f(3,6)=6,f(4,6)=6,所以 x3=0,c=6
f(4,6)=6,f(5,6)=6,所以 x4=0,c=6
f(5,6)=6≠0,所以 x5=1,
所以优化解为：x=[1,1,0,0,1]，优化值为 15。
时间复杂度分析：
证明：当重量和效益值均为整数时动态规划算法的时间复杂度为

$$O(\min\{2^n, n\sum_{1\le i\le n} p_i, nc\})$$

证明、当重量和效益值均为整数时，因为表 P(i)按 p 和 w 的增序排列，又 p 和 w 的值均为正整数，所以，其长度至多为

$$\min\{1+\sum_{i\le j\le n} p_j, c\},$$

每次产生一个表 P(i)时的计算时间和表的长度成正比，共产生 n 个表，所以总时间不超过 $O(\min\{2^n, n\sum_{1\le i\le n} p_i, nc\})$，其中，$2^n$ 是考虑到每次新产生的表顶多是前一个表长度的 2 倍。

2．解、
设 C(i)为多段图上节点 1 到节点 i 的最短路长度，设 E 为多段图的边的集合，令 A(i)={j | (j, i)∈E}，则递归式为 C(i)=min{C(j)+cji | j∈A(i) }；

就讲稿中的实例计算源到目的的最短路：自己完成。

3．设 $g(i,x)$ 表示物品 $1,\cdots,i$，背包容量 x 的 0/1 背包问题的优化效益值。

(1)试写出 $g(i,x)$ 满足的动态规划递归关系式

(2)就以下实例
n=4,c=20,w=(10,15,6,9),p=(2,5,8,1)

计算 $g(4,20)$，并回溯求出优化的物品装法。

（1）$g(i,x)=\max\{g(i-1,x),g(i-1,x-w_i)+p_i\} \quad x\ge w_i$

$g(i,x)=g(i-1,x) \quad x<w_i$

（2）对实例：n=4,c=20,w=(10,15,6,9),p=(2,5,8,1)，我们用元组法计 算 $g(4,20)$。

P(1)={(0,0)},(10,2)},Q={(15,5)}

P(2)={ (0,0)},(10,2), (15,5)},Q={(6,8),(16,10)}

P(3)={(0,0),(6,8),(16,10)}

因(6,8)+(9,1)=(15,9),效益值为 9，小于(16,10)的效益值 10。所以优化的效益值为 10。

回溯求解：g(4,20)=g(3,20)，所以 x_4=0；

g(3,20)≠g(2,20)，所以 x_3=1；

g(2,14)=g(1,14)，所以 x_2=0；

g(1,14)≠0，所以 x1=1；

直接计算得到的解：

$$g(1,x)=\begin{cases}2 & x\ge 10\\ 0 & x<10\end{cases}$$

$$g(2,x)=\begin{cases}7 & x\ge 25\\ 5 & 15\le x\le 25\\ 2 & 10\le x\le 15\\ 0 & x<10\end{cases}$$

$$g(3,x)=\begin{cases}15 & x\ge 31\\ 13 & 21\le x<31\\ 10 & 16\le x<21\\ 8 & 6\le x<16\\ 0 & x<6\end{cases}$$

4．假设一个项目的 n 个任务已按拓扑顺序排好，编号为 1 到 n；任务 1 先执行，接下来是任务 2，等等。又假定每个任务有 2 种方式完成：任务 i 按第一种方式需花费成本 $C_{i,1}$，时间 $T_{i,1}$；按第 2 种方式需成本 $C_{i,2}$，时间 $T_{i,2}$。令 cost(i,j)＝任务 1 到 i 能在 j 时间内完成的最小成本，列出 cost(i,j)满足的动态规划递归关系式。设 n＝3, T=(2,1,4;3,2,1)

C=(1,5,2;2,3,4)，t=8

试计算 cost(4,8)和优化的完成方案。

T(i,j)=任务 i 按第 j 种方式所需时间(j=1,2)

C(i,j)=任务 i 按第 j 种方式所需成本(j=1,2)

任务是拓扑排序的，必须先完成任务 1 才能完成任务 2…

要求在时间 t 之前完成所有任务且成本最小。

cost(i,j)＝任务 1 到 i 能在 j 时间内完成的最小成本。不失一般性，假定:

// T(1,1)<T(1,2)。我们有以下递归关系

cost(1,j)=∞ j<T(1,1)

=C(1,1) T(1,1)<=j<T(1,2)

=min{C(1,1),C(1,2)} T(1,2)<=j //

令 cost(0,j)= ∞ 当 j<=0；

=0 j>0

则

cost(i,j)=min{cost(i-1,j-T(i,1))+C(i,1),

cost(i-1,j-T(i,2))+C(i,2)} i>0

约定∞＋C(,)= ∞

这里约定∞表示在时间 j 内无法安排前 i 个任务。

就题中实例，计算可得：

cost(1,j)= ∞ j<2

=1 j>=2

cost(2,j)= ∞ j<3

=6 3<=j<4

=4 j>=4

cost(3,j)= ∞ j<4

cost(3,j)=10 4<=j<5

cost(3,j)=8 5<=j<8

cost(3,j)=6 j>=8

5 用动态规划设计求解最大子集和数问题的算法，即$\sum_{i=1}^{n} s_i \geq c, \quad \sum_{i \in J} s_i \leq c$, s_i 为任意正整数，J 是{1,2,⋯，n}的一个子集。要求：

（1） 列出递归关系（2）举例说明算法的执行过程。

解（1）对于下面最大子集和数问题：$\sum_{k=i}^{n} s_k \geq y, \quad \sum_{k \in J} s_k \leq y$, s_k 为任意正整数，设 f(i , y) = $\sum_{k \in J} s_k$, 其中 J 为{i,⋯，n}，约束为 y 的最大子集和数问题的解，则有以下的递归关系：$f(n,y) = \begin{cases} s_n, & y \geq s_n \\ 0, & y < s_n \end{cases}$

$$f(i,y) = \begin{cases} \max\{f(i+1,y), f(i+1,y-s_i)+s_i\}, & y \geq s_i \\ f(i+1,y), & 0 \leq y < s_i \end{cases}$$

（2）例 s = [20,18,15], c= 34，则算法的执行过程如下：

（2）例 S = [20,18,15], c= 34，则算法的执行过程如下：

$$f(3,y)=\begin{cases}15, & y \ge 15 \\ 0, & y < 15\end{cases}$$

$$f(2,y)=\begin{cases}33, & y \ge 33 \\ 18, & 18 \le y < 33 \\ 15, & 15 \le y < 18 \\ 0, & y < 15\end{cases}$$

回溯求解过程也类似：
f(1,34)=33, f(2,34)=33,所以 x1=0,
f(2,34) = 33, f(3,34)=15,所以 x2＝1，
f(3,34-18)=f(3,16)=15≠0,所以 x3＝1，
所以解的集合为 X＝[0,1,1] (J={2,3})。

矩阵乘法链见讲稿或书。