算分作业 3

24.03.11

3.3

设 $f(i,j)$ 为只考虑前 i 个货柜, 总长度不超过 j 时的最大收益, 枚举第 i 个货柜是否选取, 有转移方程

$$f(i,j)=\begin{cases}f(i-1,j), & l_i>j,\\ \max\{f(i-1,j),f(i-1,j-l_i)+v_i\}, & l_i\le j\end{cases}$$

和边界条件 $f(0,j)=0$, 最终 $f(n,D)$ 即为所求. 公式中没有指明变量的取值范围, 可以参看伪代码如下:

```
Solve
Input: l[1..n], c[1..n], D
Output: max profit
for j in [1..D]:
    f[0, j] = 0
for i in [1..n]:
    for j in [1..D]:
        f[i, j] = f[i-1, j]
        last[i, j] = last[i-1, j]
        if l[i] <= j:
            t = f[i-1, j-l[i]] + v[i]
            if t > f[i, j]:
                f[i, j] = t
                last[i, j] = i
return f[n, D]
```

这个算法的最坏时间复杂度为 $\Theta(nD)$.

3.5

设 $f(i,j)$ 为只考虑前 i 种硬币, 总价值为 j 时的最小重量, 枚举第 i 种硬币是否使用, 有转移方程

$$f(i,j)=\begin{cases}f(i-1,j), & v_i>j,\\ \min\{f(i-1,j),f(i-1,j-v_i)+w_i\}, & v_i\le j\end{cases}$$

和边界条件 $f(1,j)=w_1j$, 最终 $f(n,y)$ 即为所求. 公式中没有指明变量的取值范围, 可以参看伪代码如下:

```
Solve
Input: v[1..n], w[1..n], y
Output: min weight
for j in [1..n]:
    f[1, j] = w[1] * j
for i in [2..n]:
    for j in [1..y]:
        f[i, j] = f[i-1, j]
        last[i, j] = last[i-1, j]
        if v[i] <= j:
            t = f[i, j-v[i]] + w[i]
            if t < f[i, j]:
```

```
                f[i, j] = t
                last[i, j] = i
return f[n, y]
```

这个算法的时间复杂度为 $\Theta(ny)$.

在所给实例上的备忘录表为

i\j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	1	2	3	4	5	6	7	8	9	10	11	12
2	1	2	3	2	3	4	5	4	5	6	7	6
3	1	2	3	2	3	4	5	4	5	6	7	6
4	1	2	3	2	3	4	5	4	5	6	7	6

标记函数表为:

i\j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2	2	2	2	2	2
3	1	1	1	2	2	2	2	2	2	2	2	2
4	1	1	1	2	2	2	2	2	2	2	2	2

付钱方法为使用 3 枚第 2 种硬币.

3.7

(1)

目标函数为

$$\max_{S\subseteq\{1,\cdots,n\}}\sum_{i\in S}p_i.$$

约束条件为

$$|x_i-x_j|\ge d,\forall i,j\in S,i\ne j.$$

(2)

先将位置数组排序使 $x_1<\cdots<x_n$.

容易 $O(n)$ 地预处理出数组 y, 其中 y_i 是满足 $k<i,x_i-x_k\ge d$ 的最大 k, 即每个 x_i 原点侧相距至少为 d 的最近商店, 若不存在这样的商店, 记 $y_i=0$.

设 $f(i)$ 为只考虑前 i 个商店的最大收益, 枚举第 i 个商店是否选取, 有转移方程 $f(i)=\max\{f(i-1),f(y_i)+p_i\},1\le i\le n$ 和边界条件 $f(0)=0$, 最终 $f(n)$ 即为所求.

这个算法的最坏时间复杂度为 $\Theta(n\log n)$.

3.12

设 $f(i,j)$ 为将顶点 $i, i+1\cdots, j$ 顺序连接形成的凸多边形的最小划分总权值, 枚举 i,j 所在三角形的另一个顶点, 有转移方程

$$f(i,j)=\min_{i<k<j}\{f(i,k)+f(k,j)+d_{ik}+d_{kj}+d_{ij}\}, 2\le i+1<j\le n$$

和初始条件 $f(i,i+1)=0, 1\le i<n$, 最终 $f(1,n)$ 即为所求.

这个算法的时间复杂度是 $\Theta(n^3)$.

3.15

假定不允许加工之前剩余的任务, 不允许提前加工之后的任务.

设 $g(i,j)$ 为从第 i 天 (刚刚重启) 开始, 连续工作到第 j 天加工的总任务数, 有递推方程
$g(i,j)=g(i,j-1)+\min\{x_j, s_{j-i+1}\}, i<j$ 和边界条件 $g(i,i-1)=0$.

设 $f(i)$ 为只考虑前 i 天时的最大加工任务数, 枚举最近的重启日, 有转移方程

$$f(i)=\max\{g(1,i), \max_{1<k<i}\{f(k-1)+g(k+1,i)\}\}, 1\le i\le n$$

和边界条件 $f(0)=0$, 最终 $f(n)$ 即为所求.

这个算法的时间复杂度是 $\Theta(n^2)$.

3.17

设 $f(i,j)$ 为以 a_{ij} 为终点的路径的最小数字和, 枚举路径的最后一条边, 有转移方程

$$f(i,j)=\begin{cases} a_{ij}+f(i-1,j), & j=1, \\ a_{ij}+f(i-1,j-1), & j=i>1, \\ a_{ij}+\min\{f(i-1,j-1), f(i-1,j)\}, & 1<j<i \end{cases}$$

和边界条件 $f(0,0)=0$, 最终 $\min_{1\le j\le n} a_{nj}$ 即为所求.

这个算法的时间复杂度是 $\Theta(n^2)$.