3.3

类似于 0-1 背包问题，问题转化为

$$max \sum_{i=1}^{n} v_i x_i (x_i \text{取} 0 or 1)$$
$$\sum_{i=1}^{n} l_i x_i \leq D$$

令 $C[k,y]$ 代表只考虑前 k 个货柜，库房长度为 y 的最大收益。

递推公式有

$$C[k,y] = \begin{cases} C[k-1,y](y < l_k) \\ max(C[k-1,y], C[k-1,y-l_k]+v_k)D \geq y \geq l_k \end{cases}$$

边界条件

$$C[1,y] = \begin{cases} v_1 \qquad y \geq l_1 0 \qquad y < l_1 \end{cases}$$

伪代码:

for y<-1 to D

$C[1,y] < -v_1$

for k<-2 to n

for y<-1 to D

C[k,y] <- C[k-1,y]

hspace*1cm i[k,y] <- i[k-1,y] (标记函数)

if $y \geq l_k and C[k-1,y-l_k]+v_k > C[k-1,y]$

then $C[k,y] \leftarrow C[k-1,y-l_k]+v_k$

$i[k,y] \leftarrow k$ 最坏时间发杂度 $O(nD)$

3.6

令 F(k,y) 表示用前 k 种钱币，总钱数为 y 时可以付款。

F(k,y) = 0 or 1

递推方程为 $F(k,y) = max(F(k-1,y), F(k-1,y-v_k))$

边界条件：

$$F(1,y)=\begin{cases}1 \quad if \quad y==v_1\\ 0 \quad else\end{cases}$$

$F(k,y)==0 \quad if \quad y<0$ 标记函数则为 i(k,y)
递推方程为

$$i(k,y)=\begin{cases}k \quad if \quad F(k-1,y-v_k)==1\\ i(k-1,y)\end{cases}$$

时间复杂度为 $O(nM)$

3.12

记顶点 i，i+1,$\cdots$j 构成的凸多边形记为 A[i+1,j]，原始问题为 A[2,n] 记 A[i,j] 的最小权值为 t[i,j]，考虑 edge(i-1,j)，edge(i-1,j) 一定属于凸多边形 A[i,j] 分成的某个子三角形 ijk，则可以划分子问题为 t(i,k) 和 t(k+1,j) 有递推方程

$$t[i,j]=\begin{cases}0 \quad i==j\\ \min\limits_{i\le k<j}(t(i,j),t(i,k)+t(k+1,j)+d(i-1,j)+d(i-1,k)+d(k,j))\end{cases}$$

不考虑优化的话，时间复杂度为 $O(n^3)$

3.15

采用动态规划算法解决问题记 F[j]，表示第 1-j 天加工任务的最大数目。
w[i,j] 表示第 i-j 天连续工作的加工量。
则 $w[i+1,j]=\sum_{k=i+1}^{j}\min(x_k,s_{k-i})$
w[i,j] 满足递推方程

$$\begin{cases}w[i+1,j]=w[i+1,j-1]+\min(x_j,s_{j-i})\\ w[i+1,i+1]=\min(x_{i+1},s_1)\end{cases}$$

预处理 w[] 数组，可在 $O(n^2)$ 时间内得到。

考虑 F[j], 假设最后一天检修为第 i 天（或者没检修) 有递推方程:

$$F[j] = \max \begin{cases} \max\limits_{0<i<j}(F[i-1] + w[i+1, j]) & j > 1 \\ w[1, j] & j > 0 \end{cases}$$

特别地 F(0) = 0 时间复杂度为 $O(n^2)$
总时间复杂度为 $O(n^2)$

3.17

令 F[i,j] 表示从 a_{11}到a_{ij} 的路径上的数最小和。
则 $F[i,j] = min(F[i-1,j], F[i-1,j-1]) + a_{ij}$
特别地 $F[i,0] = +\infty, F[i,j](j > i) = +\infty$
问题即求 $\min_{1\leq j\leq n}(F[n,j])$ 确立标记函数 pos(i,j) 递推方程为

$$pos(i,j) = \begin{cases} j \quad if \quad F[i-1,j] < F[i-1,j-1] \\ j-1 \quad else \end{cases}$$

时间复杂度 $O(n^2)$