

动态规划解题总览

问题	状态	最优值(计数)	状态转移方程	初始化	最终所求
最长公共子序列(LCS)	(i,j): 考察 X 串的 i-前缀 X _i 与 Y 串的 j-前缀 Y _j	c(i,j): X _i 与 Y _j 的 LCS 长度	$c(i, j) = \begin{cases} c(i-1, j-1) + 1 & x_i = y_j \\ \max\{c(i-1, j), c(i, j-1)\} & x_i \neq y_j \end{cases}$	c(0,j)=0, j=0,..., n c(i,0)=0, i=0,..., m	c(m,n)
编辑距离	(i,j): 考察 X 串的 i-前缀 X _i 与 Y 串的 j-前缀 Y _j	d(i,j): X _i 与 Y _j 的编辑距离	$d(i, j) = \begin{cases} d(i-1, j-1) & x_i = y_j \\ \min\{d(i-1, j), d(i, j-1), d(i-1, j-1)\} & x_i \neq y_j \end{cases}$	d(0,j)=j, j=0,..., n d(i,0)=i, i=0,..., m	d(m,n)
整数划分数问题	(i,m): 限制划分元素不超过 m	q(i,m): 元素不超过 m 的划分数	$\begin{aligned} q(i, m) &= q(i, m-1) + q(i-m, m) \\ q(i, m) &= q(i, i) \quad (m > i) \end{aligned}$	q(i,1)=1 q(0,m)=1	q(n,n)
标准二维表 (方法 1)	(i,j): 第 1 行有 i 个数字, 第 2 行有 j 个数字, 包含数字 1~(i+j) 的局部表	p(i,j): 局部表的个数	$p(i, j) = \begin{cases} p(i, j-1) + p(i-1, j) & j < i \\ p(i, j-1) & j = i \end{cases}$	p(i,0)=1	p(n,n)

问题	状态	最优值(计数)	状态转移方程	初始化	最终所求
标准二维表 (方法 2)	(i,j): 第 1 行有 i 个数字, 第 2 行比第 1 行少 j 个数字的局部表	a(i,j): 把局部表填满的方式数	$a(i, j) = \begin{cases} a(i+1, j+1) + a(i, j-1) & j > 0 \\ a(i+1, j+1) & j = 0 \end{cases}$	a(n,j)=1, j=0,1,...,n	a(0,0)或 a(1,1)
防卫导弹	i: 首攻第 i 个导弹	M[i]: 首攻第 i 个导弹时的最多攻击数	$M[i] = \max_{j>i, h[j] \leq h[i]} \{M[j] + 1\}$	M[n]=1	max{M[i]}
连续邮资问题 (方法 1)	(i,v): 至多 i 张邮票, 邮资额为 v	b[i][v]: 至多 i 张邮票是否可得邮资金额为 v 的逻辑值	$b[i][v] = \bigvee_{\substack{k=1, \dots, m \\ x_k \leq v}} \{b[i-1][v-x_k]\}$	b[0][v]=0, (v>0) b[0][0]=1	首个 b[n][v]为假的金额 v 减 1
连续邮资问题 (方法 2)	v: 邮资额	p[v]: 邮资额为 v 时所需最少邮票数	$p[v] = \min_{x_k \leq v} \{p[v-x_k] + 1\}$	p[0]=0	首个 p[v]>n 的金额 v 减 1
最大子段和	j: 考察 a[1..n] 的 j-前缀	b[j]: j-前缀的最大后缀和	$b[j] = \max\{b[j-1] + a[j], a[j]\}, \quad 1 \leq j \leq n$	b[0]=0	$\max_{1 \leq j \leq n} b[j]$
独立任务最优调度问题 (方法 1)	(i,t): 安排前 i 个作业, 机器 A 的处理时间不超过 t	F[i][t]: 安排前 i 个作业, 机器 A 的处理时间不超过 t 时机器 B 的最短处理时间	$F[i][t] = \min\{F[i-1][t-a_i], F[i-1][t] + b_i\} \quad (i > 0, t > 0)$ $0 \leq i \leq n, \quad 0 \leq t \leq \min(\sum a_k, \sum b_k)$	$F[i][0] = \sum_{k=1}^i b_k$ $F[0][t] = 0$	$\min_t \{\max_t(t, F[n][t])\}$

问题	状态	最优值(计数)	状态转移方程	初始化	最终所求
独立任务最优调度问题 (方法 2*)	(i, t_1, t_2) : 考察机器 A 已被占用时间 t_1 , 机器 B 已被占用时间 t_2 时, 安排前 i 个作业	$G[i][t_1][t_2]$: 机器 A 已被占用时间 t_1 , 机器 B 已被占用时间 t_2 时, 安排前 i 个作业的最短处理时间	$G[i][t_1][t_2] = \begin{cases} \min\{G[i-1][t_1+a_i][t_2], G[i-1][t_1][t_2+b_i]\} & (i > 0) \\ t_2 + G[i][t_1-t_2][0] & (t_1 \geq t_2) \\ t_1 + G[i][t_2-t_1][0] & (t_1 \leq t_2) \end{cases}$	$G[0][0][t_2] = t_2$ $G[0][t_1][0] = t_1$	$G[n][0][0]$
分石子问题	(i, j) : 装前 i 个石子到 j 个筐, $i=1, \dots, N$; $j=1, \dots, K$	$f(i, j)$: 装前 i 个石子到 j 个筐的最优值	$f(i, j) = \min_{j-1 \leq p \leq i-1} \{ \max(f(p, j-1), \sum_{k=p+1}^i Q_k) \} \quad i=1, \dots, N; j=1, \dots, K$	$f(i, 1) = \sum_{k=1}^i Q_k$	$f(N, K)$
错位排列	i : 排列的阶	D_i : i 阶错位排列数	$D_i = (i-1)(D_{i-1} + D_{i-2}) \quad (i > 2)$ 设第 i 位数字为 j , 对数字 n 所在位置是否为 j 分情况讨论	$D_1=0$ $D_2=1$	D_n
The triangle(poj163)	(i, j) : 考察第 i 行第 j 个数开始的下行路径 ($1 \leq i \leq n$, $1 \leq j \leq i$)	$S(i, j)$: 从第 i 行第 j 个数开始的所有下行路径的最小和	$S(i, j) = a(i, j) + \min\{S(i+1, j), S(i+1, j+1)\}$ ($i = n-1, \dots, 1; j = 1, \dots, n$)	$S(n, j) = a(n, j)$ ($1 \leq j \leq n$)	$S(1, 1)$
World cup noise(poj1953)	i : 长度为 i	$S(i)$: 长度为 i 的序列种数	$S(i) = S(i-1) + S(i-2) \quad (i = 2, \dots, n)$	$S(0)=1$ $S(1)=2$	$S(n)$

问题	状态	最优值(计数)	状态转移方程	初始化	最终所求
Post office(poj1160)	(v,p): 考察前 v 个村, 限设 p 个邮局	F(v,p): 前 v 个村限设 p 个邮局的最小总距离	$F(v, p) = \min_{1 \leq k \leq v-p+1} \{F(v-k, p-1) + G(p-k+1, p)\} \quad (v > p)$ <p>G(p₁,p₂)为在 p₁~p₂ 村仅设一个邮局的最小总距离</p>	F(v,v)=0 F(v,1)=G(1,v)	F(V, P)
Alphacode(poj2033)	k: k-前缀 k=0,1,...,n	C(k): k-前缀的译码种数	$C(k) = ('1' \leq x_k \leq '9') ? C(k-1) : 0$ $+ ('10' \leq x_{k-1}x_k \leq '26') ? C(k-2) : 0 \quad (k \geq 2)$	C(1)=1 C(0)=1	C(n)
滑雪 (poj1088)	(r,c): 坐标	L(r,c): 从(r,c)开始的最长滑道长度	$L(r, c) = \max_{\substack{(r',c') \text{ 是 } (r,c) \text{ 的邻点} \\ \text{且 } h(r',c') < h(r,c)}} \{L(r', c')\} + 1$ <p>按 h 从小到大的顺序计算</p>	L(r ₀ ,c ₀)=1 (h(r ₀ ,c ₀)最小)	max L(r,c)
Palindrome (poj1159)	n: 考察子串 X _{ij} =x _i ...x _j	c(i,j): 子串 X _{ij} 需插入的最小字符数	$c(i, j) = \begin{cases} c(i+1, j-1) & x_i = x_j \\ \min\{c(i+1, j), c(i, j-1)\} + 1 & x_i \neq x_j \end{cases} \quad (i < j)$	c(i,i)=0 c(i,i-1)=0 (空串)	c(1,N)
A decorative fence (poj1037)					