



动态规划解题の 通用技巧

By zhan8855



目录

- 1 状态的设计
- 2 转移的设计
- 3 方程的特点和优化

1 状态的设计

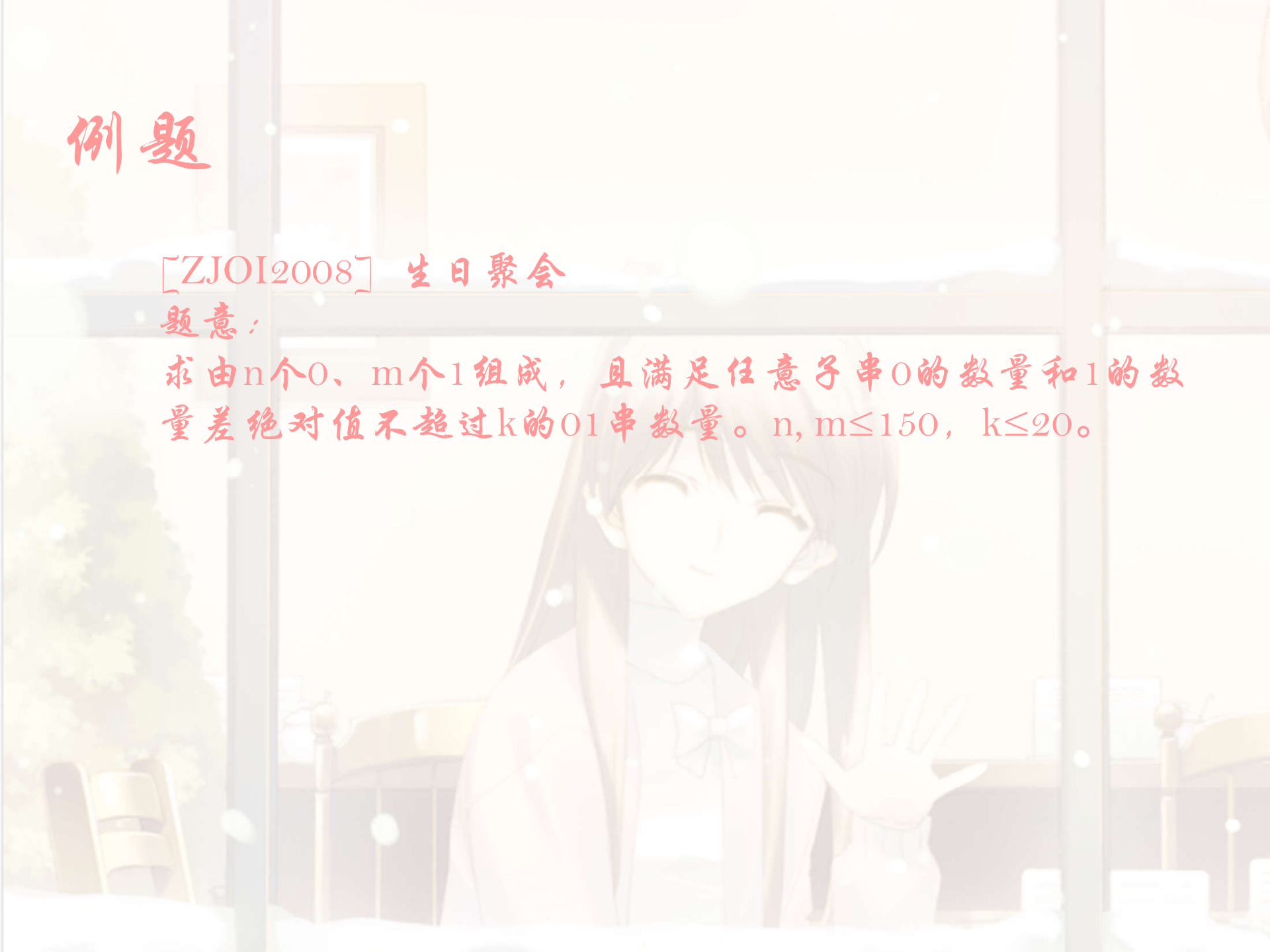
- ✓ 要尽量利用题目允许的时间和空间，尽可能清晰地实现。
- ✓ 要用尽量少的信息清晰地描述状态，求精求简。
- ✓ 要便于方程的转移。

例题

[ZJOI2008] 生日聚会

题意：

求由 n 个0、 m 个1组成，且满足任意子串0的数量和1的数量差绝对值不超过 k 的01串数量。 $n, m \leq 150, k \leq 20$ 。



例题

[SCOI2005] 最大子矩阵

题意：

在 $2 \times N$ 的矩阵中求 K 个不重叠的子矩阵，使子矩阵之和最大。 $N \leq 100, K \leq 10$ 。



2 转移的设计

- ✓ 要考虑周到，保证动态规划的正确性。
- ✓ 要充分利用状态的定义，用尽可能简单的状态和方程进行转移。
- ✓ 与状态的设计相辅相成。

例题

[Codeforces722E] Research Rover

题意：

给出一个 $N \times M$ 的方格阵，从 $(1,1)$ 出发，到 (N,M) 结束，从 (x,y) 只能走到 $(x+1,y)$ 或 $(x,y+1)$ 。方格阵上还有 K 个特殊点，初始时给出的分数 t 每经过一个特殊点就会变成 $\text{ceil}(t/2)$ 。求到 (N,M) 时得分的期望。保证 $(1,1)$ 和 (N,M) 不是特殊点。 $N, M \leq 100000$, $K \leq 2000$, $t \leq 10000000$ 。

2 转移的设计

动态规划计数去重的一般思路：

- ✓ 容斥原理
- ✓ 分类讨论
- ✓ 最小表示
- ✓ 重设状态

3 方程的特点和优化

3.1 决策点的单调性

3.2 对象的独立性

3.3 转移的单一性

3.4 方程的冗余性

3.5 预处理技巧

3.6 常数优化策略

3.1 决策点的单调性

- ✓ 如果观察状态转移方程得到决策点关于状态的单调性关系，就可以对方程进行一个比较显著的优化，通常可以降低一维时间复杂度。

例题

[USACO 2007 November Gold] Telephone Wire

题意：

有 N 根电线杆，要给相邻的两根连线。可以选择抬高其中一部分电线杆，抬高一根电线杆操作的代价为 ΔH^2 。抬高后，要给相邻的电线杆连线，连线相邻两根电线杆操作的代价为 $C \cdot \Delta H$ ，其中 C 是一个常量且已经给出。求最小总代价。

3.1 决策点的单调性

- ✓ 区间型动态规划的合并点若满足单调性，就可以由此入手优化动态规划。这种方法称为四边形不等式。
- ✓ 但是，有时候直接观察或许并不能得到结果。可以使用确定左端点、观察决策点随右端点的变化来确定思路。

3.1 决策点的单调性

- ✓ 对于较复杂的决策点满足单调性的情况，可以采用二分的方式确定每个决策点对应的决策区间。

3.1 决策点的单调性

- ✓ 斜率优化是另一种形式的决策点单调性的利用，适用于类似直线形式的动态规划方程。
- ✓ 一般思路是把问题转化为直线集合在某一点的最值，用凸壳来维护。

3.1 决策点的单调性

- ✓ 如果决策点不满足单调性或者决策点有多个，但是转移方程中的一些变量满足单调性，方程可以用高级数据结构（如线段树）来维护。

3.2 对象的独立性



例题

[Codeforces712D] Memory and Scores

题意：

两个人玩游戏，共进行 t 轮，每人每轮从 $[-k, k]$ 中选出一个数字，将其加到自己的总分中。已知两人的初始得分分别为 a 和 b ，求第一个人最后获胜的方案数。两种方案被认为是不同的，当且仅当存在其中一轮，其中一人选到的数字不同。 $a, b, t \leq 100, k \leq 1000$ 。

3.2 对象的独立性

- ✓ 对于相互独立的对象，分别计算往往比同时计算效率更优秀。

例题

[BestCoder Round #86 1004] Keep In Touch

题意：

有三个人从一张 N 个点无重边的有向无环图上的三个点出发，每单位时间，他们分别选择当前点的一条出边走下去。有向无环图点有点权，任意时刻他们所在的三个点两两点权相差不超过 K 。他们可以在任意三个点同时结束。求合法的路径总数。 $N \leq 50$ 。

3.3 转移的单一性

- ✓ 根据某些转移不变的性质，和时空允许的条件，可以使用矩阵乘法优化方程式。

3.4 方程的冗余性

- ✓ 有些方程虽然比较复杂，但是实际有效的状态和转移有限，经过挖掘，也许就可以发现优化的策略。

例题

[USACO 2009 March Gold] Cleaning Up

题意：

把长度为 N 的数列分成任意多段，每段数列的不和谐度为该段内不同数字数量的平方，总不和谐度为所有段的不和谐度之和，求总不和谐度的最小值。 $N \leq 40000$ 。

例题

[Codeforces729F] Financiers Game

题意：

两个人分别从长度为 n 的数列的两端开始取数，如果前一个人取了 k 个数，后一个人必须取 k 或 $k+1$ 个，第一个人最开始可以取1个或2个，不能操作时结束。两个人都希望自己取到的数字之和尽量大，并保持绝对理智，求最后他们取到的数字之和之差。 $n \leq 4000$ 。

3.4 方程的冗余性

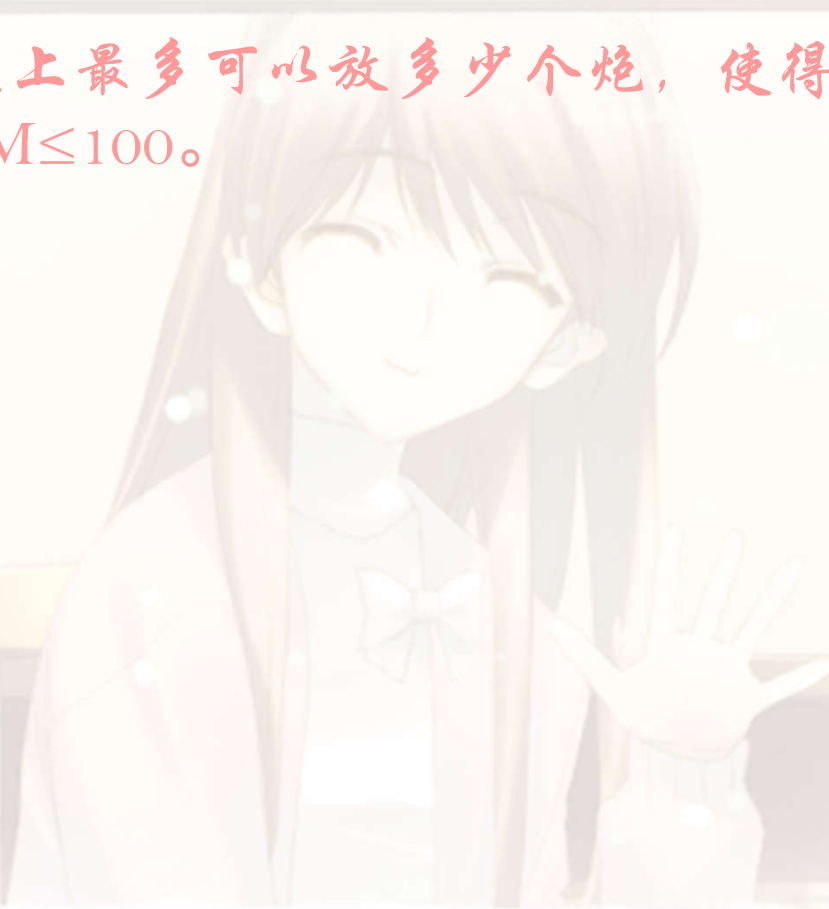
- ✓ 有些方程状态或转移虽多，但是经过分类后发现情况并不多。这时可以重新设立状态，利用状态类别进行转移。

例题

[AHOI2009] 中国象棋

题意：

求 $N \times M$ 的棋盘上最多可以放多少个炮，使得它们两两不互相攻击。 $N, M \leq 100$ 。



3.4 方程的冗余性

- ✓ 动态规划时，特别是两种同类型的状态表示，如果可以通过其中多维表示另一维，或是用一些巧妙的方法把两者结合起来，可以达到降低时间、空间、代码复杂度的效果。

例题

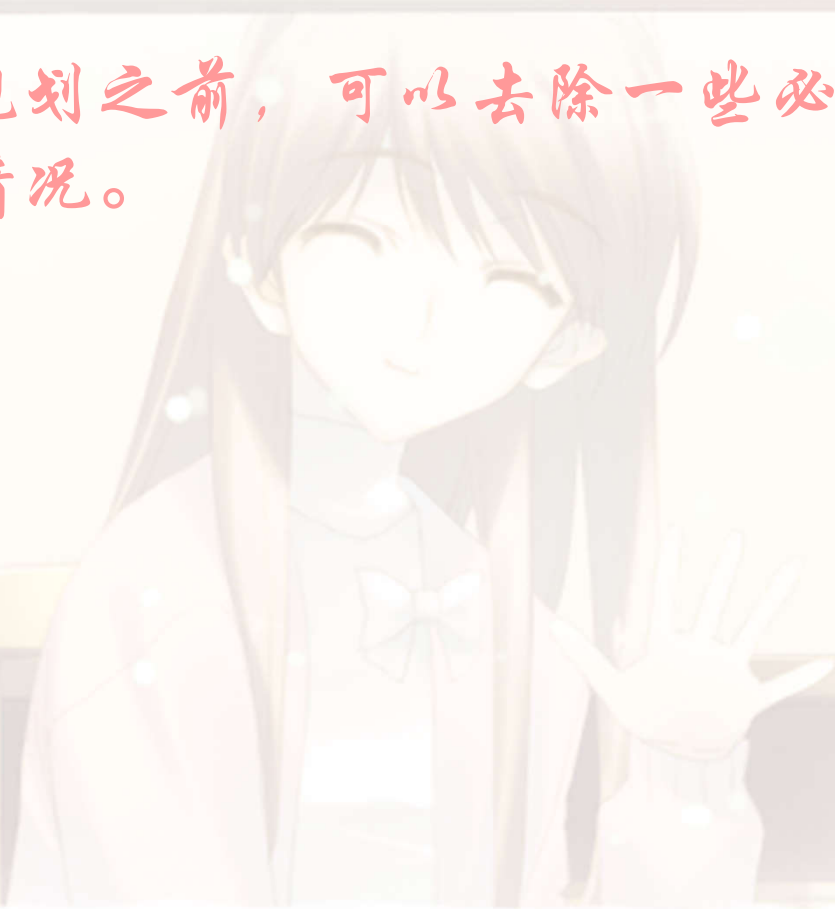
[Codeforces570E] Pig and Palindromes

题意：

给出一个 $N \times M$ 的字符矩阵，求从左上角到右下角、每次只能往右或者往下走，经过的路径上所有字符恰好为一个回文串的方案数。 $N, M \leq 500$ 。

3.5 预处理技巧

- ✓ 动态规划之前，可以去除一些必定不合法的情况。



3.5 预处理技巧

- ✓ 前缀和差分法是动态规划预处理的重要方法，能够优化区间转移的过程。

例题

[Codeforces498B] Name That Tune

题意：

用 T 秒时间按顺序听 N 首歌，第 i 首歌播放时间为 t_i 秒，且每播放一秒都会有 p_i 的概率被识别出来，跳到下一首。求听歌数量的期望。 $1 \leq N, T \leq 5000$ 。

3.5 预处理技巧

- ✓ 前驱的记录是另一种重要方法，能够使动态规划快速找到转移的状态。

例题

[BeiJing2010] 取数游戏

题意：

给出一个长度为 N 数列 A_i ，要求按顺序取出尽量多的数，并且满足相邻两个取出的数最大公约数大于 L 。 $N \leq 50000$ ， $A_i \leq 1000000$ 。

3.6 常数优化策略

- ✓ 对于常数优化空间较大的问题，合理地进行剪枝可能会对解题起到决定性作用。
- ✓ 但对于大多数问题，理论复杂度仍然是算法效率的决定性因素。

