动态规划选讲

王浏清

2024/9

动态规划

- **什么是动态规划**
  - 状态设计
  - 转移方程
- 根据状态的特点分成各种类型的 DP。
- 根据转移方程的特点进行优化。

线性 DP

- 状态的各个维度呈线性增长。
- 不一定是一维线性 DP。

[NOIP2010] 乌龟棋

- 给定一个 N 个格子的棋盘，第 i 个格子上的分数是 a_i。
- 现在有 M 张卡牌，使用第 i 张卡牌可以向前走 b_i 格，$1 \le b_i \le 4$。
- 现在给定 N, M, a, b，求最多能获得的分数。获得的分数就是走过的格子的分数之和。
- 每种卡片数量 ≤ 40。

[NOIP2010] 乌龟棋

- 观察到我们只关心当前走到了哪里，以及各种卡片还剩多少张。
- 通过剩余卡片数量可以推出使用数量，然后推出当前位置。
- 所以我们的状态只用记录 4 种卡片还剩多少张。
- 枚举最后使用的卡片类型转移。

f_{i_1, i_2, i_3, i_4}

f_{i_1+1}

[USACO20FEB] Help Yourself G

- 给定 N 个区间 $[l_i, r_i]$。
- 定义一个区间的集合 S 的代价是线段的并的连通块数。
- 对于 N 个区间的 2^N 个子集，求出它们的代价之和。
- 保证所有端点不相同。

10^6

2^i 个子集 代价之和

f_i :

$f_i \to f_{i+1}$

$f_{i+1} = f_i + 1 + dx$

[USACO20FEB] Help Yourself G

$f_3 = 8.$

$f_9 = f_3 +$

- 首先我们将所有区间按左端点排序
- 定义状态 f_i 表示前 i 条线段的 2^i 个子集的答案。
- 讨论是否选择第 i 个区间，得到方程 $f_i = f_{i-1} + f_{i-1} + 2^x$
- x 表示前 $(i-1)$ 个区间中右端点 $< l_i$ 的区间数量。

$f_i = f_{i-1} + f_{i-1} + 2^x$

x y

l r

$Ans = \left(\sum 2^x \cdot (2^y - 1)\right) + 2^n$

[USACO20FEB] Help Yourself G

- 这道题还有不使用动态规划的做法。

[NOI2009] 管道取珠

- 给定两个长度分别是 n, m 的 $0/1$ 序列。
- 现在归并这两个序列，有 $\binom{n+m}{n}$ 种方案。
- 最后得到一个长度为 $n+m$ 的序列。对于一个序列 p，记 a_p 表示有多少方案可以获得输出序列 p。
- 求 $\sum a_p^2$。
- $n, m \le 500$。

$n=2,\ m=1$ $\qquad 2^2+1^2=5$

01 $\qquad 110 \qquad 101$

1 $\qquad 2 \qquad 1$

$a_{110}=2 \qquad a_{101}=1$

$$\sum a_p = \binom{n+m}{n}$$

$$|S| = n$$

$$n^2 = (x, y),\quad x, y \in S$$

[NOI2009] 管道取珠

- 讨论平方的组合意义。
- 假设已有集合 S, $|S|^2$ 等价于二元组 (x, y) 数量，$x, y \in S$。
- 对于这道题来说，等价于方案的二元组的数量，满足两个方案的输出相同。

(x, y)　　$|S| = a_i$

[NOI2009] 管道取珠

- 讨论平方的组合意义。
- 假设已有集合 S，$|S|^2$ 等价于二元组 (x, y) 数量，$x, y \in S$。
- 对于这道题来说，等价于方案的二元组的数量，满足两个方案的输出相同。
- 因此我们可以设计状态 f_{a_1,b_1,a_2,b_2} 分别表示第一元和第二元的情况。
- 由于 $a_1 + b_1 = a_2 + b_2$，可以优化一维。
- 空间不够，需要滚动数组优化。

$u_{a_1+1} = v_{s+1-a_2}$

$u_{a_1+1} = u_{a_2+1}$

$f_{s,a_1,a_2} \to f_{s+1,a_1+1,a_2+1}$

$f_{s,a_1,a_2} \to f_{s+1,a_1+1,a_2}$

$0\sim1000$, $0\sim500$

$f_s \to f_{s+1}$

背包问题

$S: 1+2+4+\cdots 2^k + c$

-1

-2

-4

-2^k

c

2^{k+1}

$S = 2^{k_1} + 2^{k_2} + \cdots + 2^0 + c$

$\log + 1$

- 一种特殊的线性 DP。
- 体积，价值，数量。
- 最基础的三种 0/1 背包，完全背包，多重背包。
- 三种转移。

$S \uparrow$

① for $i = n \to v_i$

$f_{i-v_i} \longrightarrow f_i$

② for $i = 0 \to n-v_i$

$f_i \longrightarrow f_{i+v_i}$

[HEOI2013] EDEN 的新背包问题

- 给定 n 个物品，第 i 个物品的体积，价值，数量分别是 a_i, b_i, c_i。
- q 次询问，每次询问 d_i, e_i，表示去掉第 d_i 种物品，背包体积为 e_i 的答案。
- $n, e_i \leq 1000$，$q \leq 3 \times 10^5$。

[HEOI2013] Eden 的新背包问题

- 首先不考虑删除的物品，我们需要跑多重背包。
- 怎么删除物品？
- 背包的删除是困难的，我们可以将除去物品 d_i 的其他物品合并。
- 求出前缀背包，后缀背包，f, g。
- 合并答案 $Ans = \min\limits_{0 \le i \le e} f_{d-1,i} + g_{d+1,e-i}$

[BalticOI 2022 Day1] Uplifting Excursion

$(m,-1)$ $(m-1,-1)$ $\vee$ $+m$ T

$-m$ $-m+1$ $\dots$ 0

- 有 $2m+1$ 种物品，重量分别为 $-m,-m+1,\dots,m-1,m$。重量为 i 的物品有 a_i 个。
- 你需要拿走若干物品，使得这些物品重量之和恰好为 l。在此基础上，你需要拿尽可能多的物品。
- 问在物品重量之和恰好为 l 的基础上，你最多能拿多少物品。
- $m \le 300, a_i \le |l| \le 10^{18}$

1 2 $\dots$ m m $m-1$ $m-2$ $\dots$

1 1 1 -1 -1 -1 -1

[BalticOI 2022 Day1] Uplifting Excursion

- 观察特点，这道题要求重量恰好。
- 数量和背包容量很大，单个物品的体积很小并且有规律。
- 最重要的一点，所有物品的价值都是 1。

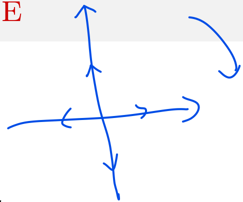
[BalticOI 2022 Day1] Uplifting Excursion

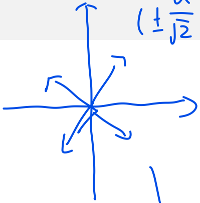
- 所有物品的价值都是 1，我们可以考虑贪心。
- 先尽量多选物品。之后要么所有物品被选完了，要么最多还剩 m 的容积。
- 最后我们用 DP 来微调最后的这点体积。
- 由于 i 和 $-i$ 不会同时出现，所以背包只用开 m^2 即可。

[BalticOI 2022 Day1] Uplifting Excursion

- 小 trick，我们可以强制选择重量为负数的物品，然后将所有重量变成正数。
- 对于贪心选择的物品，需要考虑反悔操作。所以最后跑背包时应该有四种类型的物品。

ABC221G - Jumping sequence





$(\pm\frac{d}{\sqrt{2}}, \pm\frac{d}{\sqrt{2}})$

- 给定 a, b 和一个长度为 n 的序列 d_i。
- 从 $(0,0)$ 出发，每次可以选择向上/向下/向左/向右走 d_i 的长度。
- 问是否能够走到 (a,b)。
- $n \leq 2000, |a|, |b| \leq 3*10^6$。

$(\pm d, \pm d)$

$d_1 \quad d_2, \quad ---, \quad d_n \quad = a$

ABC221G - Jumping sequence

- 通过简单归约我们知道这题一定是背包。
- 因为如果强制只能上下移动，则这题严格不弱于背包。
- 这题怎么转化为背包问题?

ABC221G - Jumping sequence

$\overbrace{0101}^{64}$

- 非常巧妙，我们将坐标系旋转 45 。
- 这样 x 轴和 y 轴独立，转化为两个一维的问题，背包解决。
- 本题 $n(|a|+|b|) \geq 3 \times 10^9$，怎么优化?
- 可以通过 bitset 优化。

$\frac{n(|a|+|b|)}{64}$

区间 DP

- 也是线性 DP 的一种
- 一般状态包括区间左右端点 l, r。

[USACO21FEB] Modern Art 3 G

① $f_{l,r} = \min_{k} \{ f_{l,k} + f_{k+1,r} \}$

② $f_{l,r} = \min_{a_l = a_r} \{ f_{l,r-1} ; f_{l+1,r} \}$

- 给定一个长度为 N 的序列，每个位置表示一种颜色。
- 每次可以将一段连续的区间染成相同的颜色。
- 染色可以覆盖。问最少需要染色多少次。
- $N \leq 300$。

1 2 1

1 2 3 2 1

2 3 2

3

1 2 3 2 1

1 1 1 1 1

1 2 2 2 1

1 2 3 2 1

[USACO21FEB] Modern Art 3 G

- 首先套路的转移，枚举区间断点 k。
- 考虑颜色覆盖操作。
- 怎么表示？可以反过来看。
- 颜色覆盖可以看成先涂上层颜色，然后向外延申下层颜色。
- 所以如果 $a_l = a_r$，$f_{l,r} = \min\{f_{l+1,r}, f_{l,r-1}\}$

[USACO21OPEN] Balanced Subsets P

- 给定一个 $N \times N$ 的方阵，每个位置为 0 或 1。
- 一个合法的连通块（四连通）需要满足：
- 1. 每个位置都是 1。
- 2. 如果 $(x_1, y), (x_2, y)$ 在连通块中，则两点之间的点也在连通块中
- 3. 如果 $(x, y_1), (x, y_2)$ 在连通块中，则两点之间的点也在连通块中
- $N \leq 150.$

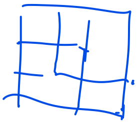
[USACO21OPEN] Balanced Subsets P

- 首先转化条件，合法连通块等价于连通块是一个“凸包”。
- 如果从上到下依次考虑每一行。则左边界和右边界都是单峰函数。
- 我们可以用 $f_{i,l,r,0/1,0/1}$ 表示考虑到第 i 行，左端点为 l，右端点为 r，左端点是否过峰，右端点是否过峰。
- 算出转移。前缀和优化。

$f_{i,l,r,0,0} \leftarrow f_{i-1,l',r',0,0}$

$l' \geq l \quad r' \geq r$

$f_{i,0,0,0/1,0/1}$



$0 \to 0,1$

$1 \to 1$

3^2

树形 DP

- 在树上跑的 DP，状态一般是节点和子树。

[CSP-S2019] 括号树

- 给定一棵树，每个节点上有一个小括号。
- 对于每个节点，求出从根节点到它对应的括号序列，有多少子串是合法括号序列。

[CSP-S2019] 括号树

- 给定一棵树，每个节点上有一个小括号。
- 对于每个节点，求出从根节点到它对应的括号序列，有多少子串是合法括号序列。

[CSP-S2019] 括号树

- 给定一棵树，每个节点上有一个小括号。
- 对于每个节点，求出从根节点到它对应的括号序列，有多少子串是合法括号序列。
- 可以先求出到某个点结束的合法括号序列数。
- 然后求树上前缀和即可。

[CSP-S2019] 括号树

- 给定一棵树，每个节点上有一个小括号。
- 对于每个节点，求出从根节点到它对应的括号序列，有多少子串是合法括号序列。
- 可以先求出到某个点结束的合法括号序列数。
- 然后求树上前缀和即可。

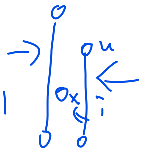
[NOI2020] 命运

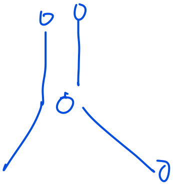
$n \le 5e5$, $m \le 10$

- 给定一棵树，和若干点对 (u, v)。点对在树上是祖先后代关系。
- 现在对于树上每条边，可以选择染色成 0 或 1。 2^{n-1}
- 求有多少方案，使得对于每一个点对对应的路径，都至少存在一条染色1边。
- 完成 $n \le 5000$ 的部分分。

x往上到 du 至少一次1

du=0



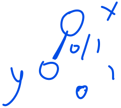


[NOI2020] 命运

- 关键信息，给出的所有路径都是祖先-后代关系的。
- 以深度为状态！
- $f_{i,j}$ 表示以 i 为根的子树内，所有未覆盖的路径中深度最深的一条深度为 j，有多少合法方案。

  起点
- j 也可以理解为在到达深度 j 之前，至少选择一条边。

  $O(n^2)$
- 类似与树形背包的合并



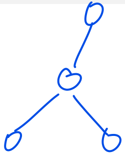
$f_{x,p}\cdot f_{y,q} \rightarrow f'_{x,\max\{p,q\}}$

$f_{x,p}\cdot f_{y,q} \xrightarrow{q\neq d_x} f'_{x,p}$

[NOI2020] 命运

- 关键信息，给出的所有路径都是祖先-后代关系的。
- 以深度为状态!
- $f_{i,j}$ 表示以 i 为根的子树内，所有未覆盖的路径中深度最深的一条深度为 j，有多少合法方案。
- j 也可以理解为在到达深度 j 之前，至少选择一条边。
- 类似与树形背包的合并

[省选联考 2022] 最大权独立集问题

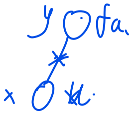


3! = 6

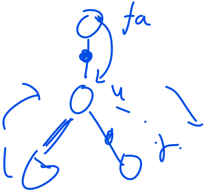
- 给定一棵 n 个节点的二叉树，第 i 号点有权值 d_i。
- 每次你可以选择一条树上存在的边，将该边删除，并将边两端的点交换。假设两端的点是 x, y，则产生 $d_x + d_y$ 的贡献。
- 求删除 $n-1$ 条边的最小代价。
- 完成 $n \leq 1000$ 的部分分。

(n-1)!

[省选联考 2022] 最大权独立集问题



- 我们可以先单独考虑一个节点。
- 由于是二叉树，所以一个点最多只有 3 条相邻边。
- 我们可以讨论这三条边的删除顺序。
- 所以我们可以设计状态 $f_{u,x,y}$ 表示以 u 为根的子树中，删除所有边。删除 u 父边时两端分别是 x, y。子树内的最小代价。
- 转移有六种，对应 6 种不同的删边顺序。
- 时空复杂度都是 $O(n^3)$。



[省选联考 2022] 最大权独立集问题

- 进一步优化？难度不在讨论范围。
- 对于 6 种不同的删边顺序，有的点会直接停留在 u，所以将状态转化为二维。
- 各种优化。dirty work。

$f_{u,x,y}$

$f_{0,1,2}$ u, x

单调队列优化

- 对线性 DP 转移的时间复杂度进行优化。
- 本质是单调队列的滑动窗口。

[USACO13OPEN] Photo G

- 给定 N 和 M 个在 $[1, N]$ 范围内的区间。
- 从 1 到 N 每个位置可以是 0 或 1。对于每个区间来说，内部恰好有一个 1。
- 求最多能有多少个 1。

$$f_i = \max_{(\quad)} \{ f_j + 1 \}$$

[USACO13OPEN] Photo G

- 条件可以转化为至少存在一个和至多存在一个。
- 至少存在一个，需要求不包含 i 的在 i 左边的左端点最大的区间。
- 至多存在一个，需要求包含 i 的左端点最小的区间。
- f_i 表示最后一个 1 在位置 i。单调队列优化。

[CSP-S2019] 划分

- 给定一个长度为 n 的序列 a。你需要将 a 划分成若干段。
- 将每一段求和得到 $s_1, s_2, \cdots$。需要满足 $s_1 \leq s_2 \leq \cdots$。
- 求最小的 $\sum s_i^2$。
- $n \leq 4 \times 10^7$。

[CSP-S2019] 划分

- 结论 $\sum s_i^2$ 最小等价于 s_n 最小。
- 证明：调整法，均值不等式。

[CSP-S2019] 划分

- 我们可以设计状态 f_i 表示以 i 结尾的最后一段最小是多少。
- $$f_i = \min_{j<i, u_i-u_j \ge f_j} \{u_i - u_j\}$$
- u 是前缀和有单调性，f 也有单调性（证明）。单调队列优化。
- 记录 f 是从哪转移的，可以得到划分方案，然后求出最终的答案。