动态规划

- 状压 DP;
- 数位 DP;
- DP 优化
  - 数据结构优化;
  - 斜率优化;
  - 决策单调性优化;
  - 四边形不等式优化。

状压 DP

OIwiki - 状压 DP

一些好题：

- 洛谷P2150 [NOI2015] 寿司晚宴
- 洛谷P3226 [HNOI2012] 集合选数
- 洛谷P4363 [九省联考 2018] 一双木棋 chess
- 洛谷P5279 [ZJOI2019] 麻将
- 洛谷P7519 [省选联考 2021 A/B 卷] 滚榜
- 洛谷P8292 [省选联考 2022] 卡牌

AT_dp_o Matching

AT_dp_o Matching

解法：

显然。

QOJ9619 乘积，欧拉函数，求和

QOJ9619 乘积，欧拉函数，求和

解法：

欧拉函数 $\varphi(n) = n\prod_i \frac{p_i-1}{p_i}$。假设我们能确定最后的质因数集合，只需要对所有可能的 n 求和。

记录考虑的素数个数 i 以及目前集合 S 入状态，容易转移。

注意到每个数只会有至多一个大于根号的素因子，我们按照其分类，这样只需要状压根号以下的因子，可以通过。

洛谷P1896 [SCOI2005] 互不侵犯

洛谷P1896 [SCOI2005] 互不侵犯

解法：

一个暴力的做法：$\mathcal{O}(4^n n^3)$。

简单优化可以做到 $\mathcal{O}(3^n n^3)$。

取出不同的状态可以做到 $\mathcal{O}(fib_n^2 n^4)$。

轮廓线 DP 可以做到 $\mathcal{O}(fib_n n^4)$。

洛谷P5074 Eat the Trees

洛谷P5074 Eat the Trees

解法：

状压轮廓线上的插头，每次转移动一个格子：

- 如果为障碍，那么不能有插头；
- 如果为空地，我们依据这个格子左上两个插头存在情况作转移。

复杂度 $\mathcal{O}(2^m nm)$。

HDU6984 Tree Planting

HDU6984 Tree Planting

解法：

我们把序列每 k 个划一排当成一个矩阵，那么限制就是不能有垂直相邻的 1，也不能有循环水平相邻的 1（这里表述不准确，大概的意思就是模拟 c_i 和 c_{i+1} 的限制，另外还有一些边角不能选择）。

通过轮廓线 DP 从上到下扫一遍就能简单做到 $\mathcal{O}(fib_k n)$，但是 k 较大时无法处理。对于这种情况，我们尝试从左到右轮廓线 DP，但第一列和最后一列有限制，我们需要提前枚举第一列的取值，再进行轮廓线 DP，复杂度 $\mathcal{O}(fib_{\frac{2n}{k}} n)$。

按照阈值分治，最终复杂度 $\mathcal{O}(fib_{\sqrt{2n}} n) = \mathcal{O}(1.618^{\sqrt{2n}} n)$。

洛谷P4590 [TJOI2018] 游园会

洛谷P4590 [TJOI2018] 游园会

解法：

考虑怎么判定，求 LCS 有一个经典的 $\mathcal{O}(nk)$ 的 DP，没有连续的 NOI 也是好判定的。

接着我们尝试暴力状压记录下求 LCS 对应的 DP 的一行。注意到一行是不降且差分不超过 1 的，我们可以状压一个 2^k 的 01 串表示，直接用目前串长，状压的 01 串，以及对 NOI 的匹配情况作为状态就能转移了，复杂度 $\mathcal{O}(2^k nk)$。

DP 套 DP 一般来说都有大部分状态没有用（以及很多状态等价类），通过 BFS 剪掉这些状态能让你代码快很多。

数位 DP

OIwiki - 数位 DP

一些好题：

- 洛谷P6371 [COCI 2006/2007 #6] V
- 洛谷P9387 [THUPC 2023 决赛] 巧克力
- CF1710C XOR Triangle

洛谷P4127 [AHOI2009] 同类分布

洛谷P4127 [AHOI2009] 同类分布

可以发现数字总和不太大，可以提前枚举，之后数位 DP 的时候记录数字和，以及对提前枚举的和取模的结果就行，复杂度 $\mathcal{O}(\log V^4)$。

CF1994G Minecraft

CF1994G Minecraft

解法：

从低到高按位考虑 x 的值，我们可以发现低位向上的进位不会超过 $n(\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\cdots)<n$ 的大小。

数位 DP 记录进位大小即可，转移的时候就枚举 x 这一个 bit 是 0 还是 1，复杂度 $\mathcal{O}(\sum nm)$。

CF1290F Making Shapes

CF1290F Making Shapes

解法：

问题容易转化为：计数序列 $\{c_i\}$ 的数量，使得对于两维，带权和为零，且两位各自的正值带权和不超过 m。

使用数位 DP，从低到高考虑，记录目前在哪一位，以及目前两维正负四个数集的进位，以及两个带权和的卡上界标记，转移直接 2^n 枚举这一层的状态即可。

复杂度 $\mathcal{O}(2^n V^4 \log m)$，其中 V 是最大进位大小，本题中为 20。

DP 优化 - 数据结构优化

OIwiki - 单调队列/单调栈优化

ARC085F NRE

ARC085F NRE

解法：

位置可以按照值分为染色扣权值和染色加权值，问题变为选择若干子区间使得并的所有位置权值和最大化。

并集不好处理，分为无交和有交。无交可以直接前缀 $\max$ 优化，有交用线段树肯定也能做，复杂度 $\mathcal{O}(n\log n)$。

有交

CF1129D Isolation

CF1129D Isolation

解法：

记 f_i 表示前 i 个数划分的答案，再动态维护一个数组 t_j 表示 $[j+1,i]$ 恰好出现一次的不同整数个数，转移则是 $f_i=\sum_{t_j\leqslant k} f_j$。

新加一个数进去就是一个 t 的区间加一，查询则是求所有 $t_j\leqslant k$ 的对应权值之和。

这个不好 polylog，我们使用序列分块，散块暴力重构，整块对所有 p 预处理每个 $t_j=p$ 的对应权值之和，这样就能方便地支持块的整体修改了（记得打一个整块标记）。

复杂度 $\mathcal{O}(n\sqrt{n})$。

CF573D Bear and Cavalry

CF573D Bear and Cavalry

解法：

假设没有自己只能骑自己的马的限制，根据排序不等式一定是两者按顺序依次匹配。加上这个限制，直觉上也不会差太多，实际上第 i 个人匹配的马与他的编号差不超过 2!

> 证明：考察一条编号差大于 2 的匹配 (x, y)，一定有至少三个匹配跨越该匹配，去除与 x, y 匹配有关的后，一定至少还剩一个，此时交换马一定能更优。

据此推广可以得到一条性质，整个序列一定由若干长度不超过 3 的"内部消化段"构成，大概可以枚举两种可能的情况，每种情况都能手玩出更好的可能。

于是我们就能写出一个 f_i 由 $f_{i-1}, f_{i-2}, f_{i-3}$ 的转移式，带修就上一个线段树维护矩阵乘法就行了，复杂度 $\mathcal{O}(n \log n)$。

DP 优化 - 单调性优化

OIwiki - 四边形不等式

OIwiki - 斜率优化

OIwiki - WQS 二分

洛谷P4767 [IOI 2000] 邮局 加强版

洛谷P4767 [IOI 2000] 邮局 加强版

解法：

首先可以列出一个 $\mathcal{O}(PV^2)$ 的 DP，只需要提前递推出每个村庄区间的代价，转移便只需要 $\mathcal{O}(V)$ 枚举决策点。

一个村庄区间的代价其实就是所有数与中位数的距离（因此也不用预处理），可以发现其满足四边形不等式，故决策点满足 $opt(i-1,j) \leqslant opt(i,j) \leqslant opt(i,j+1)$，我们从小到大枚举 i，从大到小枚举 j，并按照决策单调性的限制枚举决策点，总复杂度便降到了 $\mathcal{O}(PV)$。

这题还有很多种做法，大家可以在了解其余单调性优化方式后再来尝试。

CF833B The Bakery

CF833B The Bakery

解法：

类似上一题我们也可以列出类似的 DP，并看到对应的四边形不等式，但是求一个区间的答案还略微困难。

我们写出一维的决策单调性：$opt(i) \leqslant opt(i+1)$，接下来分治地求解各个 DP 值——每次取出区间中点，$\mathcal{O}(len)$ 地计算出它的决策点，那左侧的决策点一定在其决策点左侧，右侧也同理。

根据上面的过程，我们可以维护一个莫队状物，暴力地处理区间左右端点的位移，并实时更新不同数字的个数。可以发现分治的一层复杂度是 $\mathcal{O}(n)$ 的，故总复杂度 $\mathcal{O}(nk\log n)$。

洛谷P3515 [POI 2011] Lightning Conductor

洛谷P3515 [POI 2011] Lightning Conductor

解法：

把建筑分成前面和后面的，后面的问题是对称的，只考虑前面。可以证明四边形不等式，于是分治计算最优决策点，复杂度 $\mathcal{O}(n\log n)$。

本题还有一种线性做法，有兴趣的同学可以看题解。

洛谷P5504 [JSOI2011] 柠檬

洛谷P5504 [JSOI2011] 柠檬

解法：

首先转移的区间一定满足左端点等于右端点，转移形如：

$$\begin{aligned} f_i &= \max_{1\leqslant j\leqslant i, c_j=c_i} f_{j-1} + c_i(rk_i - rk_j + 1)^2 \\ &= c_i(rk_i+1)^2 + \max_{1\leqslant j\leqslant i, c_j=c_i} f_{j-1} + c_j rk_j^2 - 2c_i(rk_i+1)rk_j \end{aligned}$$

取 $j<k<i$，若 k 更优，则：

$$\begin{aligned} (f_{k-1} + c_k rk_k^2) - 2c_i(rk_i+1)rk_k &\geqslant (f_{j-1} + c_j rk_j^2) - 2c_i(rk_i+1)rk_j \\ \Rightarrow \frac{(f_{k-1} + c_k rk_k^2) - (f_{j-1} + c_j rk_j^2)}{rk_k - rk_j} &\geqslant 2c_i(rk_i+1) \end{aligned}$$

我们斜率优化维护的是上凸壳，目标的斜率却递增，我们应当使用单调栈来维护凸包，复杂度 $\mathcal{O}(n)$。

洛谷P5504 [JSOI2011] 柠檬

解法：

还有一种做法，我们不使用斜率优化（也就是目标斜率增加时，我们不 pop back，而是保留，询问的时候在栈上二分也可以）。

洛谷P4072 [SDOI2016] 征途

洛谷P4072 [SDOI2016] 征途

解法：

手动把方差拆开 $\mathrm{Var} = m\sum l_i^2 - (\sum l_i)^2$：

后面那坨是定值，我们也就是要最小化长度平方和。

$$f_{i,j} = \min_{k=1}^{j} f_{i-1,k-1} + (s_i - s_{k-1})^2$$

斜率优化部分与上一题类似，由于维护下凸壳，目标斜率递增，单调队列维护，复杂度 $\mathcal{O}(nm)$。（使用 wqs 二分可以做到 $\mathcal{O}(n\log m)$。）

洛谷P4655 [CEOI2017] Building Bridges

洛谷P4655 [CEOI2017] Building Bridges

解法：

先把转移式写出来，化简后 $f_i = C_i + \min_j D_j - 2h_ih_j$，按斜率优化推一下，会发现 h_i 并不递增，于是我们不能类似地通过维护单调数据结构直接计算。

一种暴力的方法是使用李超线段树，这里不赘述，复杂度 $\mathcal{O}(n\log n)$。

我们想按照斜率给直线排序依次加入，但这样编号就乱了。使用 cdq 分治对编号一维钦定顺序，此时左边算出来一个静态凸壳，右边直接单调扫描就行，复杂度同样是 $\mathcal{O}(n\log n)$。

洛谷P5308 [COCI 2018/2019 #4] Akvizna

洛谷P5308 [COCI 2018/2019 #4] Akvizna

解法：

首先写出转移式 $f_i = \max_{1 \leqslant j \leqslant i} f_{j-1} + \frac{i-j+1}{i}$，再此基础上还要带一个总轮数一维。

使用 wqs 二分来限制总轮数，然后上斜率优化优化转移，上凸壳+斜率递减，单调队列维护即可，复杂度 $\mathcal{O}(n \log k)$。

推荐阅读

- 辰星凌 - 【学习笔记】动态规划—斜率优化DP（超详细）
- Alex_Wei DP 优化方法大杂烩 I.
- Alex_Wei - DP 优化方法大杂烩 II.
- ez_lcw 决策单调性与四边形不等式
- command_block - DP的决策单调性优化总结