DP 技巧选讲

275307894a

QZEZ

2024 年 4 月 10 日

Content

披着狼皮的羊

- 将一个 DP 问题看上去的复杂度认为就是它的复杂度，这个经验在很大一部分情况下是非常对的。
- 但是仍然有一些特殊情况可以通过一定的手段分析出更低的复杂度。
- 下面我们通过一些特例来说明这个问题。

从一个经典问题开始

- HAOI2015 树上染色
- 有一棵点数为 n 的树，树边有边权。
- 给你一个在 $0 \sim n$ 之内的正整数 k，你要在这棵树中选择 k 个点，将其染成黑色，并将其他的 $n-k$ 个点染成白色。
- 将所有点染色后，你会获得黑点两两之间的距离加上白点两两之间的距离的和的收益。问收益最大值是多少。
- $n, k \leq 3000$

从一个经典问题开始

- 我们设 $dp_{i,j}$ 表示从 i 子树内选了 j 个黑色点的最大收益。直接转移是 $O(n^3)$ 的

从一个经典问题开始

- 我们设 $dp_{i,j}$ 表示从 i 子树内选了 j 个黑色点的最大收益。直接转移是 $O(n^3)$ 的
- 在合并两个子树的时候，两边的循环上界都只循环到子树大小，就是 $O(n^2)$ 的
- 这个是因为每一对点只会在 LCA 处合并一次。

经典问题 plus

- JSOI2018 潜入行动
- 有一棵 n 个点的树，你需要在树上选恰好 k 个点，使得每个点都至少有一个邻居被选择。求方案数对 10^9+7 取模。
- $n \leq 10^5, k \leq 100$

经典问题 plus

- 设 $dp_{i,j,0/1,0/1}$ 表示是否已经有儿子被选，以及自己是否被选。直接分析是 $O(nk^2)$ 的。

经典问题 plus

- 设 $dp_{i,j,0/1,0/1}$ 表示是否已经有儿子被选，以及自己是否被选。直接分析是 $O(nk^2)$ 的。
- 每次将循环上界设为 $\min(k, siz)$。假设合并 i, j 两个子树，我们分三类来讨论这个复杂度：
  - $siz_i < k, siz_j < k$：每个点在向上合并的时候最多会和 $2k$ 个 j 点进行这样的合并，总合并次数不超过 $O(nk)$。
  - $siz_i < k, siz_j \geq k$：每个点只会这样出现在 i 子树内一次，合并次数不超过 $O(nk)$。
  - $siz_i \geq k, siz_j \geq k$：这样的合并只会合并不超过 $O(\frac{n}{k})$ 次，每次 $O(k^2)$，总和不超过 $O(nk)$。

  因此总复杂度 $O(nk)$。

经典问题 plus plus

- NOI2023 模拟赛 45 夏影
- 不太记得题目了，但是应该本质相同。
- n 个点的树上每个点有一个物品，每个物品有价值和体积，你总共只能选体积之和不超过 m 的物品，且树上一条边上两端点的物品不能都选。问最大价值。
- $n \leq 100, m \leq 3000$

经典问题 plus plus

- 还是设 $dp_{i,j,0/1}$ 表示 i 子树内，选了 j 体积的物品，当前点是否选择。
- 但是好像上面的分析方法不能直接用了！

经典问题 plus plus

- 还是设 $dp_{i,j,0/1}$ 表示 i 子树内，选了 j 体积的物品，当前点是否选择。
- 但是好像上面的分析方法不能直接用了！
- 考虑一个子树，其中可以选出的体积至多为 $\min(2^{siz_i}, m)$ 个。
- 套用第二个题的三类分讨，仍然可以分析出，如果每次只转移可以组合出来的体积，复杂度是 $O(\frac{nm^2}{\log m})$ 的。
  ~~但是这个题标算是建点分树状压做到 $O(n^2 m)$~~

一个类似的题

- SNOI2024 公交线路
- 原题的前半部分我们并不需要，现在只保留后半部分：
- 有一棵点数为 n 的树，定义度数为 1 的是叶子。
- 每两个叶子之间可以连边，并覆盖树的路径上所有点。你需要对于每个 x 求出这样的连边方案数，使得每个叶子都存在一条边覆盖了 x。
- $n \leq 3000$。对 $998,244,353$ 取模。

一个类似的题

- 每个点的答案显然只和断掉这个点之后分出来的联通块大小和叶子个数有关。
- 直接 DP 转移的复杂度比较高，考虑容斥，钦定一些叶子的所有连边都不经过 x，另一些可以随便连，就可以设 $dp_{i,j}$ 表示到了第 i 个联通块，有 j 个点是可以任意连边的。直接做可以得到一个 $O(n^3)$ 的 DP。~~到这里直接 NTT 优化就可以做到 $O(n^2 \log n)$ 了，并且原题时限开了 $1.5s$ 好像是可以过的~~

一个类似的题

- 定重心为根，如果每个点只需要 DP 其子树内的联通块的话，复杂度就对了，因为这和上面的树形背包复杂度分析几乎是一样的。
- 现在只有一个子树外的联通块，发现在 DP 完前面的部分后，这个联通块不需要容斥，可以直接计算答案，于是就可以做到 $O(n^2)$ 了。

一个不是 DP 的题

~~不是 DP 的原因是因为我忘记 DP 的题在哪了~~

- CF1613F Tree Coloring
- 前面是一个简单容斥，最后是求 $\prod_{i=1}^{n}(1-d_i x)$。
- $\sum d \le n \le 10^5$。

一个不是 DP 的题

- 直接分治 NTT 是 $O(n\log^2 n)$ 的，不牛。
- 考虑按照 d_i 从大到小卷积，每次先预处理出 $(1-d_ix)^k$，然后直接将这个多项式和已经卷好的多项式卷起来。复杂度是 $O(n\log n)$ 的。

一个不是 DP 的题

- 直接分治 NTT 是 $O(n\log^2 n)$ 的，不牛。
- 考虑按照 d_i 从大到小卷积，每次先预处理出 $(1-d_ix)^k$，然后直接将这个多项式和已经卷好的多项式卷起来。复杂度是 $O(n\log n)$ 的。
- 为了证明这个复杂度，我们需要考虑历次卷积的长度和，如果能证明是 $O(n)$ 的则复杂度不超过 $O(n\log n)$。
- $(1-d_ix)^k$ 的长度和是 $O(n)$ 的，而已经卷好的多项式长度为 $\sum_{i=1}^{n}\sum_{j=1}^{n}[d_j > i] < \sum d_i \le n$，因此卷积的总长度为 $O(n)$。

总结

- 在树上的复杂度分析中，“度数之和为 $O(n)$ 级别”和“每对点合并一次是 $O(n^2)$ 的”这两个关键条件很重要。
- cxy 的集训队胡策题刻画了自顶向下的树上背包，其本质是将自下而上的树形背包过程反过来，有兴趣的同学可以去试一试。

一个我没有做出来的题

- 2021 CCPC Guilin K.Tax
- 给你一张 n 个点，m 条边的图，每条边长度为 1，并有一个颜色 c_i。
- 你要从 1 出发前往 t，你需要走最短路径，并且，如果你当前第 k 次经过颜色 c 的边，你需要支付 kw_c 的费用。
- 求从 1 走到每个点的最小费用。
- $n \leq 50, m \leq \frac{n(n-1)}{2}$

一个我没有做出来的题

- 看到这个题没有任何思路，这个贡献难以计算，即使使用我们后面要讲的组合意义方法优化，也是 $O(3^n)$ 的。
- 但是写个暴力发现直接过了！

一个我没有做出来的题

- 看到这个题没有任何思路，这个贡献难以计算，即使使用我们后面要讲的组合意义方法优化，也是 $O(3^n)$ 的。
- 但是写个暴力发现直接过了！
- 我们将这个图按照从 1 开始的最短路分层，设每一层的层数为 d，则复杂度的上界不会超过 $\prod d$。
- 同时，有 $\sum d = n$，因此当每个 d 都是 3 的时候，复杂度取到最大，为 $O(3^{\frac{n}{3}})$，可以通过。

一个知道了哪里的题

- PA2019 Desant
- 给定一个排列 A，请对所有 $k = 1 \sim n$，求出 A 的所有长度为 k 的子序列最小的顺序对数，以及顺序对数达到最小值的长度为 k 的子序列个数。
- $n \leq 40$

一个知道了哪里的题

- 考虑按照值从小到大加入，并决定这个数是否被放入子序列内。
- 如果这个数被放入了，那么对于这个数的贡献是它前面已经放入的数的个数。
- 注意到如果还有 k 个位置没有确定是否取，我们只关心这 k 个位置之间 1 的个数，将状态数缩减到这样就足以通过。

一个知道了哪里的题

- 考虑按照值从小到大加入，并决定这个数是否被放入子序列内。
- 如果这个数被放入了，那么对于这个数的贡献是它前面已经放入的数的个数。
- 注意到如果还有 k 个位置没有确定是否取，我们只关心这 k 个位置之间 1 的个数，将状态数缩减到这样就足以通过。
- 如果两个位置之间距离 k，则这两个位置之间的状态只有 k 种，即 1 至多有 $k-1$ 个，因此可以用类似上一题的思想证明其状态数是 $O(3^{\frac{n}{3}})$ 的。

一个奇奇怪怪的题

- 2021 CCPC Guilin L.Wiring Engineering
- 在平面直角坐标系下，有 n 个点在第一象限，第 i 个点坐标为 $(i,1)$，另有 n 个点在第四象限，第 i 个点坐标为 $(i,-1)$。
- 现在要在第一象限的点和第四象限的点之间连边，如果第一象限的 i 点与第四象限的 j 点有连边，那么会获得 $w_{i,j}$ 的收益。如果至少有 1 条边与某个点相连，则要付出 a_i 的代价。
- 要求边之间不能相交。设 $f(a,b,c,d)$ 表示只在第一象限 $[a,b]$ 区间与第四象限 $[c,d]$ 之间连边的最大收益。有 q 个询问，求 $f(a_i,b_i,c_i,d_i)$。
- $n \le 500, q \le 3 \times 10^5$。

一个奇奇怪怪的题

- 直接 DP 可以设 $f_{a,b,c,d,0/1,0/1}$ 表示只在第一象限的 $[a,b]$ 区间内和第四象限的 $[c,d]$ 区间内连边的最大收益，以及 b 和 d 是否选择。

一个奇奇怪怪的题

- 直接 DP 可以设 $f_{a,b,c,d,0/1,0/1}$ 表示只在第一象限的 $[a,b]$ 区间内和第四象限的 $[c,d]$ 区间内连边的最大收益，以及 b 和 d 是否选择。
- 考虑分治，假设当前分治区间 $[l,r]$，取中点 mid 并考虑 $[a_i,b_i]$ 跨过 mid 的答案。枚举 $[a_i,mid]$ 在第四象限匹配到哪里，预处理从 mid 开始和结尾的 DP 数组即可。
- 直接对第一维分治是 $O(n^3\log n)$ 的，但是如果两维轮换划分，根据主定理有 $T(n)=4T(n/2)+O(n^3)$ 仍然是 $O(n^3)$ 的，总复杂度 $O(n^3+qn)$。
- 另一个类似的题是 ZJOI2016 旅行者
- ~~另一个用这个方法理论能过的题是PA2020 Tekstówka~~

优化转移

- 上文所述的分析时间复杂度，已经从侧面给我们提供了一种优化 DP 的方法：对 DP 进行一定量的剪枝，然后分析出更优的复杂度。
- 接下来我们介绍几种基于加速转移与计算的优化方法。

另一个经典问题

- LG P1654 OSU!
- 有一个长度为 n 的序列，每个序列有 p_i 的概率为 1，每段极长连续的长度为 X 的 1 可以贡献 X^3 的权值，求最后的权值期望。
- $n \leq 10^5$

另一个经典问题

- 这题有一个利用期望的线性性拆 $(X+1)^3$ 的方法，但是和接下来要讲的本质相同。
- 我们考虑使用 X^3 的一个组合意义：X^3 相当于在长度为 X 的段中选择三个互相区分的 1 的方案数。
- 则我们可以设 $dp_{i,j}$ 表示 DP 到第 i 个数，当前选了 j 个 1 的方案数，即可做到 $O(n)$。
- NFLSOJ P5224. 达拉然的废墟 也同样用了这个方法优化。

这意味着什么？

- 这显然给了我们一种以 $O(k^2)$ 的方法维护权值为次数为 k 的多项式的做法。
- 但能不能再给力一点呢？

所衍生出来的一个问题

- NOI2009 管道取珠
- 有两个栈，长度分别为 n, m。
- 每次操作可以把其中一个栈的栈顶拿出来放到序列末尾。设 a_S 表示最终序列为 S 的方案数，求 $\sum a_S^2$。
- $n \leq 500$。

所衍生出来的一个问题

- 仿照上面的方法考虑这个问题，将平方的组合意义转化成选择两个最终序列同样为 S 的方案数。
- 因为要求的是对所有 S 求和，这相当于选两个最终序列相同的操作序列，则设 $dp_{l,r,x,y}$ 表示第一个序列两个栈分别拿了 l, r 个，第二个序列两个栈分别拿了 x, y 个，转移只需要满足当前选的相等就行。
- 进一步的，有 $i + j = l + r$，因此复杂度是 $O(n^3)$ 的。

这又意味着什么？

- 我们将方案数的平方转化成选两个最终状态一样的操作序列，这给了我们一种“复读”思路，也即在 DP 状态中重复写若干个状态，这样就达到了计算若干次方的问题。
- 你说的都对，但是这种题都是题目里一个平方写给你的，太明显了，能不能再给力一点呢？

所衍生出来的又一个问题

- AGC039F Min Product Sum
- 有一个大小为 $N \times M$ 的矩阵。矩阵中每个数的取值都是 $[1, K]$。
- 对于一个矩阵，定义函数 $f(x, y)$ 为：第 x 行和第 y 列的一共 $N + M - 1$ 个数中的最小值。
- 对于一个矩阵，定义其权值为 $\prod_{x=1}^{N} \prod_{y=1}^{M} f(x, y)$。
- 你需要求出，对于所有 K^{NM} 种矩阵，每个矩阵的权值和对 D 取模的结果。
- $1 \leq N, M, K \leq 100$，$10^8 \leq D \leq 10^9$，保证 D 为质数。

所衍生出来的又一个问题

- 这个题和前面所讲的有什么关系？

所衍生出来的又一个问题

- 这个题和前面所讲的有什么关系？
- 我们考虑对矩阵二元组 (A, B) 计数，其中 A 是题目中描述的矩阵，要求 $B_{x,y} \in [1, f(x, y)]$。显然这和原问题等价。
- $B_{x,y} \in [1, f(x, y)]$ 这个限制相当难看，不妨改写成 B 在 x 行的最大值小于等于 A 在 x 行最小值，并且 B 在 x 列最大值小于等于 A 在 x 列最小值。

所衍生出来的又一个问题

- 这个题和前面所讲的有什么关系？
- 我们考虑对矩阵二元组 (A, B) 计数，其中 A 是题目中描述的矩阵，要求 $B_{x,y} \in [1, f(x, y)]$。显然这和原问题等价。
- $B_{x,y} \in [1, f(x, y)]$ 这个限制相当难看，不妨改写成 B 在 x 行的最大值小于等于 A 在 x 行最小值，并且 B 在 x 列最大值小于等于 A 在 x 列最小值。
- 之后的转移略微繁琐，在此略去。但是问题已经可以在 $O(n^4)$ 复杂度内解决。

最后的最后

- 最后这个题解释了这个思想到底是什么：用对于元组的计数来代替乘法，来让限制更容易维护，从而达到了优化的目的。
- 同时，将平方一般化为乘法也让这个技巧的实用性更强。
~~你有没有发现讲到这里已经有点跑题了~~。

一个经典问题

- 给定一张 n 个点 m 条边的有向图，边的长度为 1，你需要计算这样的点对 (x, y) 数量，满足：
- 存在一个点 z，使得存在一条 $x \to z$ 的路径长度是一条 $y \to z$ 的路径长度的两倍。
- $n, m \leq 3 \times 10^3$。

一个经典问题

- 考虑设 $dp_{i,j}$ 表示 (i,j) 点对是否满足条件。
- 初始时有 $dp_{x,x}=1$，每次转移将 i 往反边走两步，j 往反边走一步，走到的点就是可以转移到的点。

一个经典问题

- 考虑设 $dp_{i,j}$ 表示 (i,j) 点对是否满足条件。
- 初始时有 $dp_{x,x}=1$，每次转移将 i 往反边走两步，j 往反边走一步，走到的点就是可以转移到的点。
- 优化考虑分步转移，先走 i 的第一步，然后走 i 的第二步，最后走 j 的一步，这样复杂度就是 $O(nm)$ 的。

这意味着什么

- 我们感觉看似什么都没有做——只是给转移分了个步，转移的复杂度就降低了。

这意味着什么

- 我们感觉看似什么都没有做——只是给转移分了个步，转移的复杂度就降低了。
- 我认为这个优化的本质还是合并了状态，状态在走完每一步之后相同的部分被合并了，因此达到了优化复杂度的目的。

这意味着什么

- 我们感觉看似什么都没有做——只是给转移分了个步，转移的复杂度就降低了。
- 我认为这个优化的本质还是合并了状态，状态在走完每一步之后相同的部分被合并了，因此达到了优化复杂度的目的。
- 这个优化一般最后一步用，都几乎大同小异，观察到转移步与步之间相对独立就 OK 了。~~但是不要滥用，有血的教训~~。
- 接下来讲一个不是那么明显的分步转移题。

一个不是很显然的题

- AGC039F Min Product Sum
- 有一个大小为 $N \times M$ 的矩阵。矩阵中每个数的取值都是 $[1, K]$。
- 对于一个矩阵，定义函数 $f(x, y)$ 为：第 x 行和第 y 列的一共 $N + M - 1$ 个数中的最小值。
- 对于一个矩阵，定义其权值为 $\prod_{x=1}^{N} \prod_{y=1}^{M} f(x, y)$。
- 你需要求出，对于所有 K^{NM} 种矩阵，每个矩阵的权值和对 D 取模的结果。
- $1 \leq N, M, K \leq 100$，$10^8 \leq D \leq 10^9$，保证 D 为质数。
- ~~这个题怎么又出现了一次。~~

一个不是很显然的题

- 我们不转化了，我们直接暴力 DP ！
- 从小到大枚举 k，设 $f_{i,j}$ 表示当前已经确定了 i 行 j 列的最小值，并且剩下的部分已经选好的方案数。直接转移或者容斥转移至少要枚举最小值为当前值有多少个，是 $O(n^2)$ 的。
- 我们的目标自然是分步转移做到 $O(n)$ 转移，但是这两维的增量之间对权值的贡献并不是非常独立。

一个不是很显然的题

- 我们需要让它们独立一点，为此，我们进行容斥。但是，如果对两维同时容斥，又会变得复杂，所以我们只对一维容斥。
- 不妨假设对行容斥，假设我们枚举在当前的 i, j 的基础上分别增加了 x 行 y 列最小值为 z，但是有 z 行没有 k 这个值。

一个不是很显然的题

- 我们需要让它们独立一点，为此，我们进行容斥。但是，如果对两维同时容斥，又会变得复杂，所以我们只对一维容斥。
- 不妨假设对行容斥，假设我们枚举在当前的 i, j 的基础上分别增加了 x 行 y 列最小值为 z，但是有 z 行没有 k 这个值。
- 需要给这个东西编一个转移顺序使得可以分步转移。首先我们转移 $x - z$，这些行没有限制，可以任意转移。
- 然后转移 y，因此最后 z 行不包含 k，因此现在转移的 y 列每列都要包含一个 k，这个方案数也是容易计算的。
- 最后转移 z，只需要计算不包含 k 的方案数即可。
- 因此，我们借助容斥完成了分步转移，时间复杂度 $O(n^4)$。

以有涯随无涯

- DP 的技巧浩如烟海，更何况我才疏学浅，不可能穷尽，这里只是挑选了几种我比较熟悉的来和大家分享，更多的技巧还是要从平日的题目中总结、获得，这样才能厚积薄发（~~典型的卷批思想~~）。~~当然某些天赋哥可以现场发明~~。
- 一个感触比较深的例子是”对 0 度点容斥”这个 DP 技巧今年省选的时候直接让我免于退役，~~当然这就是纯运气了~~。
- 有什么问题欢迎大家提出！

Thanks for listening!