动态规划选讲

Harry27182

目录

题意

题目大意

给定一个 n 个点 m 条边的 DAG，你可以给每个点染一个 $1 \sim k$ 的颜色，共有 k^n 中染色方案。对于一种染色方案，设第 i 种颜色的长度为 l 的链有 cnt_i 条，我们定义这种方案的权值为 $\sum_{i=1}^{k} cnt_i^2$。求所有方案的权值和。
$n \leq 300, l \leq 20$。

问题转化

- cnt_i^2 这个形式显然需要一些组合意义的转化。

问题转化

- cnt_i^2 这个形式显然需要一些组合意义的转化。
- 考虑转化为选择两条长度为 l 的颜色为 i 的链的方案数。

问题转化

- cnt_i^2 这个形式显然需要一些组合意义的转化。
- 考虑转化为选择两条长度为 l 的颜色为 i 的链的方案数。
- 拆贡献，假设两条链重合了 c 个点，那么贡献系数为 $k^{n-2l+c+1}$。问题转化为给定 c，求选出两条长度为 l 且颜色相同的链，恰好有 c 个点相交的方案数。

问题转化

- cnt_i^2 这个形式显然需要一些组合意义的转化。
- 考虑转化为选择两条长度为 l 的颜色为 i 的链的方案数。
- 拆贡献，假设两条链重合了 c 个点，那么贡献系数为 $k^{n-2l+c+1}$。问题转化为给定 c，求选出两条长度为 l 且颜色相同的链，恰好有 c 个点相交的方案数。
- 把恰好二项式反演掉，设 f_c 表示恰好 c 个点相交的方案数，显然只需计算 g_c 表示钦定 c 个点相交的方案数。

设计 dp

- 令 dp_{u,i,l_1,l_2} 表示上一个相交位置为 u，钦定了 i 个相交位置，两条链长度为 l_1,l_2 的方案数。

设计 dp

- 令 dp_{u,i,l_1,l_2} 表示上一个相交位置为 u，钦定了 i 个相交位置，两条链长度为 l_1,l_2 的方案数。
- 转移到 $dp_{v,i+1,l_1+t_1,l_2+t_2}$，预处理 $h_{u,v,i}$ 表示 $u \to v$ 长度为 i 的路径数，$s_{u,i},t_{u,i}$ 分别表示起点或终点为 u，长度为 i 的路径数即可转移。

设计 dp

- 令 dp_{u,i,l_1,l_2} 表示上一个相交位置为 u，钦定了 i 个相交位置，两条链长度为 l_1,l_2 的方案数。
- 转移到 $dp_{v,i+1,l_1+t_1,l_2+t_2}$，预处理 $h_{u,v,i}$ 表示 $u \to v$ 长度为 i 的路径数，$s_{u,i},t_{u,i}$ 分别表示起点或终点为 u，长度为 i 的路径数即可转移。
- 复杂度 $O(n^2l^5+n^3l)$。

优化

- 显然可以分别枚举 t_1, t_2，复杂度优化为 $O(n^2 l^4 + n^3 l)$。

优化

- 显然可以分别枚举 t_1, t_2，复杂度优化为 $O(n^2l^4 + n^3l)$。
- 进一步的，我们其实不关心每个 f_c，我们只关心 $\sum_c k^{n-2l+c+1} f_c$，考虑统一计算贡献系数。

优化

- 显然可以分别枚举 t_1, t_2，复杂度优化为 $O(n^2l^4 + n^3l)$。
- 进一步的，我们其实不关心每个 f_c，我们只关心 $\sum_c k^{n-2l+c+1} f_c$，考虑统一计算贡献系数。
- 推一推式子，$ans = \sum_c k^{n-2l+c+1} f_c$。

优化

- 显然可以分别枚举 t_1, t_2，复杂度优化为 $O(n^2l^4 + n^3l)$。
- 进一步的，我们其实不关心每个 f_c，我们只关心 $\sum_c k^{n-2l+c+1} f_c$，考虑统一计算贡献系数。
- 推一推式子，$ans = \sum_c k^{n-2l+c+1} f_c$。
- $ans = k^{n-2l+1} \sum_{i=0}^{l} \sum_{j=i}^{l} k^i (-1)^{j-i} g_j$

优化

- 显然可以分别枚举 t_1, t_2，复杂度优化为 $O(n^2l^4 + n^3l)$。
- 进一步的，我们其实不关心每个 f_c，我们只关心 $\sum_c k^{n-2l+c+1} f_c$，考虑统一计算贡献系数。
- 推一推式子，$ans = \sum_c k^{n-2l+c+1} f_c$。
- $ans = k^{n-2l+1} \sum_{i=0}^{l} \sum_{j=i}^{l} k^i (-1)^{j-i} g_j$
- 即可得到 $ans = k^{n-2l+1} \sum_{j=0}^{l} (k-1)^j g_j$。

优化

- 显然可以分别枚举 t_1, t_2，复杂度优化为 $O(n^2l^4+n^3l)$。
- 进一步的，我们其实不关心每个 f_c，我们只关心 $\sum_c k^{n-2l+c+1} f_c$，考虑统一计算贡献系数。
- 推一推式子，$ans=\sum_c k^{n-2l+c+1} f_c$。
- $ans=k^{n-2l+1}\sum_{i=0}^{l}\sum_{j=i}^{l} k^i(-1)^{j-i} g_j$
- 即可得到 $ans=k^{n-2l+1}\sum_{j=0}^{l}(k-1)^j g_j$。
- 所以每次转移的时候乘上 $k-1$ 作为系数即可。复杂度 $O(n^2l^3+n^3l)$。

题意

题目大意

有 m 个长度为 n 的排列，其中共有 q 个位置的值已经确定，其余位置未确定。求所有本质不同的排列组对应的 $\prod_{i=1}^{n}(\min_{j=1}^{m} p_{j,i})$ 之和。对 998244353 取模。两组排列 P,Q 本质不同，当且仅当存在 i,j 使得 $P_{i,j} \neq Q_{i,j}$。保证至少存在一种合法方案。

$n \leq 50, q \leq 10, m \leq 998244353$

$q = 0$

- 当 $q = 0$ 时，这是一个经典问题，扫值域，设 $dp_{i,j}$ 表示考虑了值域 $1 \sim i$，有 j 列被填过数的方案数。

$q = 0$

- 当 $q = 0$ 时，这是一个经典问题，扫值域，设 $dp_{i,j}$ 表示考虑了值域 $1 \sim i$，有 j 列被填过数的方案数。
- 转移考虑枚举 k 表示新填入 k 列，从 $dp_{i-1,j}$ 转移到 $dp_{i,j+k}$。

$q = 0$

- 当 $q = 0$ 时，这是一个经典问题，扫值域，设 $dp_{i,j}$ 表示考虑了值域 $1 \sim i$，有 j 列被填过数的方案数。
- 转移考虑枚举 k 表示新填入 k 列，从 $dp_{i-1,j}$ 转移到 $dp_{i,j+k}$。
- 容斥这 k 列有多少列没填入元素，转移系数为 $\binom{n-j}{k}\sum_{p=0}^{k}(-1)^p\binom{k}{p}(j+k-i+1-p)^m$。

$q=0$

- 当 $q=0$ 时，这是一个经典问题，扫值域，设 $dp_{i,j}$ 表示考虑了值域 $1\sim i$，有 j 列被填过数的方案数。
- 转移考虑枚举 k 表示新填入 k 列，从 $dp_{i-1,j}$ 转移到 $dp_{i,j+k}$。
- 容斥这 k 列有多少列没填入元素，转移系数为 $\binom{n-j}{k}\sum_{p=0}^{k}(-1)^p\binom{k}{p}(j+k-i+1-p)^m)$。
- 复杂度 $O(n^4)$。预处理和 $j-i,k$ 相关的容斥系数即可做到 $O(n^3)$。

$q \leq 5$

- 记录有关键点的列为关键列，其他为非关键列。

$q \leq 5$

- 记录有关键点的列为关键列，其他为非关键列。
- 容易想到将关键列是否被填过数也计入状态，设 $dp_{i,j,S}$ 表示考虑了值域 $1 \sim i$，当前填了 j 列，其中包含 S 中的关键列的方案数。

$q \leq 5$

- 记录有关键点的列为关键列，其他为非关键列。
- 容易想到将关键列是否被填过数也计入状态，设 $dp_{i,j,S}$ 表示考虑了值域 $1 \sim i$，当前填了 j 列，其中包含 S 中的关键列的方案数。
- 考虑特殊位置对贡献系数的影响。这种影响分为两部分，第一部分是对于当前 i 的特殊位置，他们的方案是唯一的，只能是给定的数。

$q \leq 5$

- 记录有关键点的列为关键列，其他为非关键列。
- 容易想到将关键列是否被填过数也计入状态，设 $dp_{i,j,S}$ 表示考虑了值域 $1 \sim i$，当前填了 j 列，其中包含 S 中的关键列的方案数。
- 考虑特殊位置对贡献系数的影响。这种影响分为两部分，第一部分是对于当前 i 的特殊位置，他们的方案是唯一的，只能是给定的数。
- 第二部分是对于一行中值比 i 大的特殊位置，对应的位置是不能填数的，需要空出来留给对应的数去填。

$q \leq 5$

- 记录有关键点的列为关键列，其他为非关键列。
- 容易想到将关键列是否被填过数也计入状态，设 $dp_{i,j,S}$ 表示考虑了值域 $1 \sim i$，当前填了 j 列，其中包含 S 中的关键列的方案数。
- 考虑特殊位置对贡献系数的影响。这种影响分为两部分，第一部分是对于当前 i 的特殊位置，他们的方案是唯一的，只能是给定的数。
- 第二部分是对于一行中值比 i 大的特殊位置，对应的位置是不能填数的，需要空出来留给对应的数去填。
- 同样进行容斥并统计方案数即可，记得加入特殊位置的影响。

$q \le 5$

- 记录有关键点的列为关键列，其他为非关键列。
- 容易想到将关键列是否被填过数也计入状态，设 $dp_{i,j,S}$ 表示考虑了值域 $1 \sim i$，当前填了 j 列，其中包含 S 中的关键列的方案数。
- 考虑特殊位置对贡献系数的影响。这种影响分为两部分，第一部分是对于当前 i 的特殊位置，他们的方案是唯一的，只能是给定的数。
- 第二部分是对于一行中值比 i 大的特殊位置，对应的位置是不能填数的，需要空出来留给对应的数去填。
- 同样进行容斥并统计方案数即可，记得加入特殊位置的影响。
- 复杂度 $O(n^4 4^q q)$。

正解

- 计算贡献的时候容斥有点过于菜了，我们进行一个转换。

正解

- 计算贡献的时候容斥有点过于菜了，我们进行一个转换。
- 原问题等价于对于所有序列 $x_1 \sim x_n$，求满足 $\forall i, \min_k\{p_{k,i}\} \geq x_i$ 的 p 排列组的方案数之和。

正解

- 计算贡献的时候容斥有点过于菜了，我们进行一个转换。
- 原问题等价于对于所有序列 $x_1 \sim x_n$，求满足 $\forall i, \min_k\{p_{k,i}\} \geq x_i$ 的 p 排列组的方案数之和。
- 设 $dp_{i,j,S}$ 表示当前 x_i 序列填完了 $1 \sim i$ 的数，填了 j 个位置，其中关键列对应的位置填完了 S 集合。

正解

- 计算贡献的时候容斥有点过于菜了，我们进行一个转换。
- 原问题等价于对于所有序列 $x_1 \sim x_n$，求满足 $\forall i, \min_k\{p_{k,i}\} \geq x_i$ 的 p 排列组的方案数之和。
- 设 $dp_{i,j,S}$ 表示当前 x_i 序列填完了 $1 \sim i$ 的数，填了 j 个位置，其中关键列对应的位置填完了 S 集合。
- 对于非关键行，其方案数为将 x_i 排序后 $\prod_{i=1}^{n}(i - x_i + 1)$。

正解

- 计算贡献的时候容斥有点过于菜了，我们进行一个转换。
- 原问题等价于对于所有序列 $x_1 \sim x_n$，求满足 $\forall i, \min_k\{p_{k,i}\} \geq x_i$ 的 p 排列组的方案数之和。
- 设 $dp_{i,j,S}$ 表示当前 x_i 序列填完了 $1 \sim i$ 的数，填了 j 个位置，其中关键列对应的位置填完了 S 集合。
- 对于非关键行，其方案数为将 x_i 排序后 $\prod_{i=1}^{n}(i - x_i + 1)$。
- 记录 $num_{s,p,i}$ 表示考虑 s 集合内的特殊列，p 这一行值 $\geq i$ 的限制个数。

正解

- 如果这一个位置存在限制，那么这里方案数唯一，否则方案数为 $i-x_i+1-num_{*,*,i}$。

正解

- 如果这一个位置存在限制，那么这里方案数唯一，否则方案数为 $i - x_i + 1 - num_{*,*,i}$。
- 转移系数是可以在转移的过程中快速计算的，关键行特殊处理即可。

正解

- 如果这一个位置存在限制，那么这里方案数唯一，否则方案数为 $i - x_i + 1 - num_{*,*,i}$。
- 转移系数是可以在转移的过程中快速计算的，关键行特殊处理即可。
- 复杂度 $O(n^3 3^q q)$。

正解

- 如果这一个位置存在限制，那么这里方案数唯一，否则方案数为 $i - x_i + 1 - num_{*,*,i}$。
- 转移系数是可以在转移的过程中快速计算的，关键行特殊处理即可。
- 复杂度 $O(n^3 3^q q)$。
- 发现关键列和非关键列的转移相互独立，分开转移复杂度 $O(n^2 3^q q + n^3 2^q q)$。

1. luogu P11292
2. qoj9611
3. luogu P10181
4. luogu P10717
5. qoj7769

题意

题目大意

给定一个长度为 n 的序列，每个位置有颜色 a_i。定义 $f(l,r)$ 表示区间 $[l,r]$ 的不同颜色数。给定 m 次询问 x_i,k_i，求将前缀 $[1,x_i]$ 划分为 k_i 段区间每一段最大 f 之和。
$n \leq 10^5, m \leq 10^6$

暴力

- 显然有 $dp_{i,j} = \max_{k<i}\{dp_{k,j-1} + f(k+1, i)\}$。

暴力

- 显然有 $dp_{i,j} = \max_{k<i}\{dp_{k,j-1} + f(k+1, i)\}$。
- 上述 dp 可以用线段树优化转移，复杂度 $O(n^2 \log n)$。

暴力

- 显然有 $dp_{i,j} = \max_{k<i}\{dp_{k,j-1} + f(k+1, i)\}$。
- 上述 dp 可以用线段树优化转移，复杂度 $O(n^2 \log n)$。
- 感觉一下 dp 数组关于 j 是凸的，如果是单次询问就可以 wqs 二分，复杂度 $O(n \log^2 n)$。

优化

- 如果 $a_i \le 30$，那么 wqs 二分的斜率一定是 ≤ 30 的。所以可以暴力求出每个 ≤ 30 的斜率对应的 dp 数组，扫一遍求答案即可。

优化

- 如果 $a_i \le 30$，那么 wqs 二分的斜率一定是 ≤ 30 的。所以可以暴力求出每个 ≤ 30 的斜率对应的 dp 数组，扫一遍求答案即可。
- 记录斜率为 c，注意到当 c 很大的时候，切点横坐标就会很小。设阈值 B，对于 $k \le B$，可以暴力预处理 dp 数组。

优化

- 如果 $a_i \le 30$，那么 wqs 二分的斜率一定是 ≤ 30 的。所以可以暴力求出每个 ≤ 30 的斜率对应的 dp 数组，扫一遍求答案即可。
- 记录斜率为 c，注意到当 c 很大的时候，切点横坐标就会很小。设阈值 B，对于 $k \le B$，可以暴力预处理 dp 数组。
- 对于 $k > B$，此时一定有 $c < \frac{n}{B}$，所以可以预处理所有 $c < \frac{n}{B}$ 的结果。对于询问，离线下来扫一遍即可。

优化

- 如果 $a_i \le 30$，那么 wqs 二分的斜率一定是 ≤ 30 的。所以可以暴力求出每个 ≤ 30 的斜率对应的 dp 数组，扫一遍求答案即可。
- 记录斜率为 c，注意到当 c 很大的时候，切点横坐标就会很小。设阈值 B，对于 $k \le B$，可以暴力预处理 dp 数组。
- 对于 $k > B$，此时一定有 $c < \frac{n}{B}$，所以可以预处理所有 $c < \frac{n}{B}$ 的结果。对于询问，离线下来扫一遍即可。
- 取 $B = \sqrt{n}$，复杂度 $O(n\sqrt{n}\log n + m)$。

正解

- 考虑我们线段树在支持哪些操作，在后缀插入一个数，后缀加一，查询全局最大值。

正解

- 考虑我们线段树在支持哪些操作，在后缀插入一个数，后缀加一，查询全局最大值。
- 维护一个单调递减的单调栈，第一个操作直接插入栈尾，第三个操作直接查询栈顶的值即可。

正解

- 考虑我们线段树在支持哪些操作，在后缀插入一个数，后缀加一，查询全局最大值。
- 维护一个单调递减的单调栈，第一个操作直接插入栈尾，第三个操作直接查询栈顶的值即可。
- 对于第二个操作，首先找到对应位置，可以用序列并查集维护。然后会弹出前面的若干数并打上 $+1$ 标记。

正解

- 考虑我们线段树在支持哪些操作，在后缀插入一个数，后缀加一，查询全局最大值。
- 维护一个单调递减的单调栈，第一个操作直接插入栈尾，第三个操作直接查询栈顶的值即可。
- 对于第二个操作，首先找到对应位置，可以用序列并查集维护。然后会弹出前面的若干数并打上 $+1$ 标记。
- 用链表维护单调栈，相邻两项之间维护差分数组即可。

正解

- 考虑我们线段树在支持哪些操作，在后缀插入一个数，后缀加一，查询全局最大值。
- 维护一个单调递减的单调栈，第一个操作直接插入栈尾，第三个操作直接查询栈顶的值即可。
- 对于第二个操作，首先找到对应位置，可以用序列并查集维护。然后会弹出前面的若干数并打上 $+1$ 标记。
- 用链表维护单调栈，相邻两项之间维护差分数组即可。
- 复杂度 $O(n\sqrt{n}\alpha(n)+m)$。

1. luogu P11292
2. qoj9611
3. luogu P10181
4. luogu P10717
5. qoj7769

题意

题目大意

给定一棵 n 个点的树，每个节点有一个灯泡，进行 k 次闪灯操作。第 i 个点的灯泡有 $p_{i,j}$ 的概率收到第 j 次闪灯操作。如果一个灯泡在两个进行闪灯操作的灯泡的最短路径上，那么它也会进行闪灯操作。

定义一个灯泡的美丽度为 $w_i = a_{i,S}$，S 为执行闪灯操作的集合。

求 $\prod_i w_i$ 的期望。

$n \le 100, k \le 8$。

k=1

- 对于 $k=1$ 的情况，考虑一个点是否能被选中，很容易想到记录以下三种子树状态：

k=1

- 对于 $k=1$ 的情况，考虑一个点是否能被选中，很容易想到记录以下三种子树状态：
- 0 表示子树内没有节点执行闪灯操作，1 表示子树内有节点执行闪灯操作且钦定子树外有节点执行闪灯操作，2 表示子树内有节点执行闪灯操作且钦定子树外无节点执行闪灯操作。

k=1

- 对于 $k=1$ 的情况，考虑一个点是否能被选中，很容易想到记录以下三种子树状态：
- 0 表示子树内没有节点执行闪灯操作，1 表示子树内有节点执行闪灯操作且钦定子树外有节点执行闪灯操作，2 表示子树内有节点执行闪灯操作且钦定子树外无节点执行闪灯操作。
- 不难发现本质不同的状态只有这三种。

k=1

- 对于 $k=1$ 的情况，考虑一个点是否能被选中，很容易想到记录以下三种子树状态：
- 0 表示子树内没有节点执行闪灯操作，1 表示子树内有节点执行闪灯操作且钦定子树外有节点执行闪灯操作，2 表示子树内有节点执行闪灯操作且钦定子树外无节点执行闪灯操作。
- 不难发现本质不同的状态只有这三种。
- 考虑转移，发现大部分转移都是显然的，只有 $1 \to 2$ 的转移无法做到只和 u, v 有关，记录辅助状态 3 表示至少出现过两个 1 即可转移。

k=1

- 对于 $k=1$ 的情况，考虑一个点是否能被选中，很容易想到记录以下三种子树状态：
- 0 表示子树内没有节点执行闪灯操作，1 表示子树内有节点执行闪灯操作且钦定子树外有节点执行闪灯操作，2 表示子树内有节点执行闪灯操作且钦定子树外无节点执行闪灯操作。
- 不难发现本质不同的状态只有这三种。
- 考虑转移，发现大部分转移都是显然的，只有 $1 \to 2$ 的转移无法做到只和 u, v 有关，记录辅助状态 3 表示至少出现过两个 1 即可转移。
- 最后去考虑 u 是否被选择，转移也是简单的。复杂度 $O(nk)$。

暴力

- 显然可以对每次闪灯操作状态压缩，上面有效的转移共有 8 个，复杂度 $O(n(8^k + 7^k))$。

暴力

- 显然可以对每次闪灯操作状态压缩，上面有效的转移共有 8 个，复杂度 $O(n(8^k+7^k))$。
- 对于合并根节点状态的部分，注意到这是一个类似于高维后缀和的操作，类比 FMT 进行操作即可做到 $O(k4^k)$。

暴力

- 显然可以对每次闪灯操作状态压缩，上面有效的转移共有 8 个，复杂度 $O(n(8^k+7^k))$。
- 对于合并根节点状态的部分，注意到这是一个类似于高维后缀和的操作，类比 FMT 进行操作即可做到 $O(k4^k)$。
- 复杂度 $O(n(8^k+k4^k))$。

正解

- 对于 $O(8^k)$ 的部分，注意到如果没有 3 这个辅助状态，转移的复杂度为 $O(5^k)$。

正解

- 对于 $O(8^k)$ 的部分，注意到如果没有 3 这个辅助状态，转移的复杂度为 $O(5^k)$。
- 对于结果为 3 的部分，可以用结果为 0/1/3 的部分减去结果为 0/1 的部分。

正解

- 对于 $O(8^k)$ 的部分，注意到如果没有 3 这个辅助状态，转移的复杂度为 $O(5^k)$。
- 对于结果为 3 的部分，可以用结果为 0/1/3 的部分减去结果为 0/1 的部分。
- 发现前者就是 $(0/1, 0/1)$，所以可以把 0/1 状态都加到 3 上面，然后只进行 0/1/2 之间的转移和 $(3,3) \to 3$ 的转移即可。

正解

- 对于 $O(8^k)$ 的部分，注意到如果没有 3 这个辅助状态，转移的复杂度为 $O(5^k)$。
- 对于结果为 3 的部分，可以用结果为 $0/1/3$ 的部分减去结果为 $0/1$ 的部分。
- 发现前者就是 $(0/1, 0/1)$，所以可以把 $0/1$ 状态都加到 3 上面，然后只进行 $0/1/2$ 之间的转移和 $(3,3) \to 3$ 的转移即可。
- 这样得到的 3 是 $0/1/3$，减去 $0/1$ 的部分即可。

正解

- 对于 $O(8^k)$ 的部分，注意到如果没有 3 这个辅助状态，转移的复杂度为 $O(5^k)$。
- 对于结果为 3 的部分，可以用结果为 0/1/3 的部分减去结果为 0/1 的部分。
- 发现前者就是 $(0/1, 0/1)$，所以可以把 0/1 状态都加到 3 上面，然后只进行 0/1/2 之间的转移和 $(3,3) \to 3$ 的转移即可。
- 这样得到的 3 是 0/1/3，减去 0/1 的部分即可。
- 前后两部分都可以用 FWT/IFWT 实现，复杂度 $O(n(6^k + k4^k))$。

1. luogu P11292
2. qoj9611
3. luogu P10181
4. luogu P10717
5. qoj7769

题意

题目大意

给定一颗 n 个节点的树，边权为 0/1/?。你要给每条未被确定边权的边确定一个 0 或者 1 的边权，然后从树上取出若干条有向路径，使得这些链两两之间满足边不相交。然后你会把这些路径插入一颗 0/1-Trie，你希望最大化这颗 0/1-Trie 上的节点数。
在原题中你还需要构造方案，但这并没有什么意义，所以只求出最大值即可。
$n \le 18$。

$w \in \{0, 1\}$

- 我们把所有的路径提取下来，按照字典序排序，不难发现最后的答案就是选取路径的 len 之和减去相邻两两间 LCP 长度。

$w \in \{0, 1\}$

- 我们把所有的路径提取下来，按照字典序排序，不难发现最后的答案就是选取路径的 len 之和减去相邻两两间 LCP 长度。
- 令 $dp_{s,i}$ 表示上次选择 i 条路径，选择了 s 集合的点的最优解。枚举下一条路径转移，复杂度 $O(2^n n^4)$。

$w \in \{0,1\}$

- 我们把所有的路径提取下来，按照字典序排序，不难发现最后的答案就是选取路径的 len 之和减去相邻两两间 LCP 长度。
- 令 $dp_{s,i}$ 表示上次选择 i 条路径，选择了 s 集合的点的最优解。枚举下一条路径转移，复杂度 $O(2^n n^4)$。
- 考虑优化，注意到 $lcp(i,j) = \min_{k=i}^{j-1}\{lcp(k,k+1)\}$，设 $dp_{s,i,j}$ 表示考虑到第 i 条路径，选择了 s 集合内的点，上次选择到现在的 $\min lcp = j$ 的最优解。复杂度 $O(2^n n^3)$。

正解

- 对于 $w \in \{0, 1, ?\}$ 的情况，不同的路径可能会有 $O(2^n)$ 条，复杂度为 $4^n\text{poly}(n)$。

正解

- 对于 $w \in \{0,1,?\}$ 的情况，不同的路径可能会有 $O(2^n)$ 条，复杂度为 $4^n\text{poly}(n)$。
- 但这个真的能跑满吗？

正解

- 对于 $w \in \{0,1,?\}$ 的情况，不同的路径可能会有 $O(2^n)$ 条，复杂度为 $4^n\text{poly}(n)$。
- 但这个真的能跑满吗？
- 注意到对于一条长度为 x 的路径，它最多有 2^x 种不同情况，他每次更新的时候显然只会更新他的边集的超集，这个大小是 2^{n-x-1} 的，所以每条路径更新的复杂度是 $O(2^n)$ 的。

正解

- 对于 $w \in \{0,1,?\}$ 的情况，不同的路径可能会有 $O(2^n)$ 条，复杂度为 $4^n \text{poly}(n)$。
- 但这个真的能跑满吗？
- 注意到对于一条长度为 x 的路径，它最多有 2^x 种不同情况，他每次更新的时候显然只会更新他的边集的超集，这个大小是 2^{n-x-1} 的，所以每条路径更新的复杂度是 $O(2^n)$ 的。
- 使用滚动数组来避免继承的复杂度，转移暴力枚举超集，这两部分复杂度均为 $O(2^n n^3)$。

正解

- 对于 $w \in \{0,1,?\}$ 的情况，不同的路径可能会有 $O(2^n)$ 条，复杂度为 $4^n\mathrm{poly}(n)$。
- 但这个真的能跑满吗？
- 注意到对于一条长度为 x 的路径，它最多有 2^x 种不同情况，他每次更新的时候显然只会更新他的边集的超集，这个大小是 2^{n-x-1} 的，所以每条路径更新的复杂度是 $O(2^n)$ 的。
- 使用滚动数组来避免继承的复杂度，转移暴力枚举超集，这两部分复杂度均为 $O(2^n n^3)$。
- 瓶颈在于和 lcp 取 min 的下传部分，我们暴力枚举上一个转移串在哪里的时候有可能需要下传，暴力下传那个串的超集。每个串只会被考虑 $O(n)$ 次，复杂度 $O(2^n n^3)$。

另一种做法

- 感谢 yyc 同学提出以下做法：

另一种做法

- 感谢 yyc 同学提出以下做法：
- 注意到 $dp_{s,i,j}$ 状态的 i 变成了 $O(2^n)$ 级别的，值域为 $O(n)$ 级别的，所以可以考虑交换值域定义域。

另一种做法

- 感谢 yyc 同学提出以下做法:
- 注意到 $dp_{s,i,j}$ 状态的 i 变成了 $O(2^n)$ 级别的,值域为 $O(n)$ 级别的,所以可以考虑交换值域定义域。
- 显然 ans 相同的时候 i 越小越好。所以令 $dp_{s,i}$ 表示选了 s 集合,答案为 i 的最大字典序路径最小值。

另一种做法

- 感谢 yyc 同学提出以下做法：
- 注意到 $dp_{s,i,j}$ 状态的 i 变成了 $O(2^n)$ 级别的，值域为 $O(n)$ 级别的，所以可以考虑交换值域定义域。
- 显然 ans 相同的时候 i 越小越好。所以令 $dp_{s,i}$ 表示选了 s 集合，答案为 i 的最大字典序路径最小值。
- 枚举下一条路径的起点终点，对答案的增量进行转移，这样就需要 LCP 在某个值以内，可以通过一些预处理 $O(1)$ 算出答案。

另一种做法

- 感谢 yyc 同学提出以下做法：
- 注意到 $dp_{s,i,j}$ 状态的 i 变成了 $O(2^n)$ 级别的，值域为 $O(n)$ 级别的，所以可以考虑交换值域定义域。
- 显然 ans 相同的时候 i 越小越好。所以令 $dp_{s,i}$ 表示选了 s 集合，答案为 i 的最大字典序路径最小值。
- 枚举下一条路径的起点终点，对答案的增量进行转移，这样就需要 LCP 在某个值以内，可以通过一些预处理 $O(1)$ 算出答案。
- 复杂度 $O(2^n n^4)$，剪剪枝可能能更快。