状压 DP 和一些别的题目选讲

nantf

2023 年 9 月 22 日

碎碎念。

【清华集训 2014】主旋律

- 给定一个 n 个点 m 条边的有向图，无重边无自环。
- 问所有 2^m 个边的子集中，有多少个满足仅保留这个子集后整个图强连通。对大质数取模。
- $1 \leq n \leq 15$。

【清华集训 2014】主旋律

- 强连通这个条件的整体性太强了。正难则反，非强连通的话可以分成若干个强连通分量，变成子问题。

【清华集训 2014】主旋律

- 强连通这个条件的整体性太强了。正难则反，非强连通的话可以分成若干个强连通分量，变成子问题。
- 设 f_S 表示只考虑 S 这个点集，整个图强连通的方案数。数不强联通的方案数，DAG 可以采用拓扑排序，所以枚举一个子集 T 表示 T 这些点形成若干个强连通分量，且缩点后入度为 0，然后转成 $S-T$ 的子问题。

【清华集训 2014】主旋律

- 强连通这个条件的整体性太强了。正难则反，非强连通的话可以分成若干个强连通分量，变成子问题。
- 设 f_S 表示只考虑 S 这个点集，整个图强连通的方案数。数不强联通的方案数，DAG 可以采用拓扑排序，所以枚举一个子集 T 表示 T 这些点形成若干个强连通分量，且缩点后入度为 0，然后转成 $S-T$ 的子问题。
- 入度为 0 的点是必须两两不连边的。所以再设一个 g_S 表示把 S 分成若干个无边相连的强连通图的方案数。转移可以直接枚举编号最小点所在分量子集，$O(3^n)$。

【清华集训 2014】主旋律

- 这样会把一个 DAG 数重，因为我们很难做到每次枚举的 T 就恰好是所有入度为 0 的分量。所以改用容斥，枚举 T 表示只要求 T 这里面的入度为 0。容斥系数就是 $(-1)^{|T|-1}$。

【清华集训 2014】主旋律

- 这样会把一个 DAG 数重，因为我们很难做到每次枚举的 T 就恰好是所有入度为 0 的分量。所以改用容斥，枚举 T 表示只要求 T 这里面的入度为 0。容斥系数就是 $(-1)^{|T|-1}$。
- 预处理一些类似于点和子集之间边数的信息，时间复杂度 $O(3^n)$。

- 这样会把一个 DAG 数重，因为我们很难做到每次枚举的 T 就恰好是所有入度为 0 的分量。所以改用容斥，枚举 T 表示只要求 T 这里面的入度为 0。容斥系数就是 $(-1)^{|T|-1}$。
- 预处理一些类似于点和子集之间边数的信息，时间复杂度 $O(3^n)$。
- 或许可以用一些集合幂级数的东西做到 $O(\mathrm{soft}(2^n))$，但是和这次定位不符~~而且我懒得细想~~就不讲了。

IOI2023 集训队互测 R3T1 整数

- 给定长度为 2^n 的 01 序列 $\{b_i\}$。保证 $b_0 = 1$。
- 我们称一个长度为 n 的序列 $\{a_i\}$ 是好的，当且仅当在每一个二进制位 k 下，所有 a_i 这一位的取值拼接成的 n 位二进制数 s 满足 $b_s = 1$。也即

$$\forall k \in \mathbb{N}, b_{\sum_{i=1}^{n} 2^{i-1} \lfloor \frac{a_i}{2^k} \rfloor} = 1$$

- 给定长度为 n 的序列 $\{r_i\}$。问有多少个满足 $a_i \le r_i$ 的序列 $\{a_i\}$ 是好的。对大质数取模。
- $1 \le n \le 18, 0 \le r_i < 2^{60}$。

IOI2023 集训队互测 R3T1 整数

- 直接使用数位 dp，从低位到高位，令 $f_{i,S}$ 表示填了最低的 i 位，目前 S 这些位置超过了 r 的限制。初始状态 $f_{-1,\varnothing}=1$，答案为 $f_{59,\varnothing}$。

IOI2023 集训队互测 R3T1 整数

- 直接使用数位 dp，从低位到高位，令 $f_{i,S}$ 表示填了最低的 i 位，目前 S 这些位置超过了 r 的限制。初始状态 $f_{-1,\varnothing}=1$，答案为 $f_{59,\varnothing}$。
- 转移就是看第 i 位填了什么（注意 b 的限制）：
  - 若 a_j 这一位填了 0，而 r_j 这一位是 1，那么就一定没有超过限制。
  - 若 a_j 这一位填了 1，而 r_j 这一位是 0，那么就一定超过了限制。
  - 否则是否超过限制与之前相同。

IOI2023 集训队互测 R3T1 整数

- 直接使用数位 dp，从低位到高位，令 $f_{i,S}$ 表示填了最低的 i 位，目前 S 这些位置超过了 r 的限制。初始状态 $f_{-1,\varnothing}=1$，答案为 $f_{59,\varnothing}$。
- 转移就是看第 i 位填了什么（注意 b 的限制）：
  - 若 a_j 这一位填了 0，而 r_j 这一位是 1，那么就一定没有超过限制。
  - 若 a_j 这一位填了 1，而 r_j 这一位是 0，那么就一定超过了限制。
  - 否则是否超过限制与之前相同。
- 发现实际上是若 r_j 这一位是 1，则将 S 的第 j 位与 a_j 取与，若 r_j 这一位是 0 则为取或。所以可以直接看成 f_{i-1} 与 b 的某种卷积，使用每一维各自前缀和/后缀和的 FMT 就可以了。

IOI2023 集训队互测 R3T1 整数

- 直接使用数位 dp，从低位到高位，令 $f_{i,S}$ 表示填了最低的 i 位，目前 S 这些位置超过了 r 的限制。初始状态 $f_{-1,\varnothing}=1$，答案为 $f_{59,\varnothing}$。
- 转移就是看第 i 位填了什么（注意 b 的限制）：
    - 若 a_j 这一位填了 0，而 r_j 这一位是 1，那么就一定没有超过限制。
    - 若 a_j 这一位填了 1，而 r_j 这一位是 0，那么就一定超过了限制。
    - 否则是否超过限制与之前相同。
- 发现实际上是若 r_j 这一位是 1，则将 S 的第 j 位与 a_j 取与，若 r_j 这一位是 0 则为取或。所以可以直接看成 f_{i-1} 与 b 的某种卷积，使用每一维各自前缀和/后缀和的 FMT 就可以了。
- 时间复杂度 $O(2^n n\log r)$。

IOI2023 集训队互测 R15T1 翻修道路

- 给定一个 n 个点 m 条边的有向图，每条边有边权 a_i。你可以对每条边进行升级，第 i 条边升级后的边权会变为 b_i。保证 $1 \le b_i \le a_i$。
- 给定 k 个关键点的编号，对于每个 $x \in [0, m]$，问在升级至多 x 条边的情况下，从 1 号点到 k 个关键点的最短路的最大值最小是多少。
- $1 \le n, m \le 100, 1 \le k \le 8$。

IOI2023 集训队互测 R15T1 翻修道路

- 一个解下从 1 到 k 个关键点的最短路的并，形成了以 1 为根的外向树。在这个基础上可以考虑 dp。

IOI2023 集训队互测 R15T1 翻修道路

- 一个解下从 1 到 k 个关键点的最短路的并，形成了以 1 为根的外向树。在这个基础上可以考虑 dp。
- 令 $f_{u,i,S}$ 表示从 u 开始，升级至多 x 条边，走到 S 这个集合中最短路的最大值最小。对应了外向树上以 u 为根的子树，钦定这个子树里含有 S 这些点。这个 dp 形式上与斯坦纳树较为类似，但其实自然不少（

IOI2023 集训队互测 R15T1 翻修道路

- 一个解下从 1 到 k 个关键点的最短路的并，形成了以 1 为根的外向树。在这个基础上可以考虑 dp。
- 令 $f_{u,i,S}$ 表示从 u 开始，升级至多 x 条边，走到 S 这个集合中最短路的最大值最小。对应了外向树上以 u 为根的子树，钦定这个子树里含有 S 这些点。这个 dp 形式上与斯坦纳树较为类似，但其实自然不少（
- 转移也与斯坦纳树类似，分为两种。第一种是枚举一条树边，分类讨论升级或不升级，从 (v,i,S) 转移到 $(u,i/i+1,S/S\cup\{u\})$。这一部分是 i,S 两维分层的分层图最短路，故复杂度为 $O(m2^k\times m\log n)$。

- 第二种是合并一个点的两棵子树。从 $(u, i, S), (u, j, T)$ 转移到 $(u, i+j, S \cup T)$。朴素转移的时间复杂度为 $O(n \times m^2 \times 3^k)$，不能通过。

IOI2023 集训队互测 R15T1 翻修道路

- 第二种是合并一个点的两棵子树。从 $(u,i,S),(u,j,T)$ 转移到 $(u,i+j,S\cup T)$。朴素转移的时间复杂度为 $O(n\times m^2\times 3^k)$，不能通过。
- 注意转移方式是两边的 max 的最小值，所以可以在第二维用双指针优化，也即不妨只考虑 $f_{u,j,T}\le f_{u,i,S}$ 的转移，那么固定 u,S,T 时，对于每个 i 找到第一个满足的 j 就可以了，这个 j 随 i 递增不降。这一部分复杂度 $O(n\times m\times 3^k)$，可以通过。

「THUSCH 2017」巧克力

- 给定一个 $n \times m$ 的表格，每个格子有一个权值 $a_{i,j}$ 和颜色 $c_{i,j}$。$c_{i,j} = -1$ 表示这个格子不可用。
- 问最小的四连通块大小，使得该连通块内没有不可用的格子，且包含至少 k 种不同的颜色。在大小最小的前提下，求出连通块内权值中位数的最小值。
- T 组数据，$1 \le T \le 5, 1 \le n \times m \le 233, 1 \le k \le 5$。

「THUSCH 2017」巧克力

- 首先考虑颜色一共只有 k 种的情况。

「THUSCH 2017」巧克力

- 首先考虑颜色一共只有 k 种的情况。
- 网格图四连通块加上数据范围可以联想到插头 dp。毛估估一下发现根本没有过的可能，而且考虑到我很友好所以不可能会讲插头 dp 题，所以直接放弃这个想法。

「THUSCH 2017」巧克力

- 首先考虑颜色一共只有 k 种的情况。
- 网格图四连通块加上数据范围可以联想到插头 dp。毛估估一下发现根本没有过的可能，而且考虑到我很友好所以不可能会讲插头 dp 题，所以直接放弃这个想法。
- 仍然考虑斯坦纳树。直接套用最典型的状态 $f_{u,S}$ 表示目前的连通块以 u 为根，包含了 S 这些颜色，连通块的最小大小。在按照 S 分层后每一层只需要 bfs，所以复杂度为 $O(3^k nm)$。

「THUSCH 2017」巧克力

- 首先考虑颜色一共只有 k 种的情况。
- 网格图四连通块加上数据范围可以联想到插头 dp。毛估估一下发现根本没有过的可能，而且考虑到我很友好所以不可能会讲插头 dp 题，所以直接放弃这个想法。
- 仍然考虑斯坦纳树。直接套用最典型的状态 $f_{u,S}$ 表示目前的连通块以 u 为根，包含了 S 这些颜色，连通块的最小大小。在按照 S 分层后每一层只需要 bfs，所以复杂度为 $O(3^k nm)$。
- 要求中位数最小值也不难，二分中位数 x，那么要求连通块内 $\leq x$ 的减去 $> x$ 的尽可能多，所以把状态改成一个二元组，第二维存这个就行了。复杂度为 $O(3^k nm \log nm)$。

「THUSCH 2017」巧克力

- 回到颜色数任意多的情况。直接 $O((nm)^k)$ 枚举最优解出现的 k 种颜色显然是不可取的。我们发现只保留 k 种颜色浪费了大量信息，可以转为考虑把所有颜色分成 k 个组若干次，使得对于任意 k 种颜色，在至少一次被分在了互不相同的组里。

「THUSCH 2017」巧克力

- 回到颜色数任意多的情况。直接 $O((nm)^k)$ 枚举最优解出现的 k 种颜色显然是不可取的。我们发现只保留 k 种颜色浪费了大量信息，可以转为考虑把所有颜色分成 k 个组若干次，使得对于任意 k 种颜色，在至少一次被分在了互不相同的组里。
- 但是构造好烦，所以直接随机分组好了。最优解中出现的 k 种颜色被分开的概率是 $k!/k^k$，本题数据范围下高达 3.84%，所以大概跑个一百多次就可以了。

「THUSCH 2017」巧克力

- 回到颜色数任意多的情况。直接 $O((nm)^k)$ 枚举最优解出现的 k 种颜色显然是不可取的。我们发现只保留 k 种颜色浪费了大量信息，可以转为考虑把所有颜色分成 k 个组若干次，使得对于任意 k 种颜色，在至少一次被分在了互不相同的组里。
- 但是构造好烦，所以直接随机分组好了。最优解中出现的 k 种颜色被分开的概率是 $k!/k^k$，本题数据范围下高达 3.84%，所以大概跑个一百多次就可以了。
- 时间复杂度 $O(T \times 3^k nm \times \log nm \times \frac{k^k}{k!})$。

IOI2023 集训队互测 R8T1 环覆盖

- 给定一个 n 个点 m 条边的简单无向图。
- 对每个 $i \in [0, m]$，计算保留恰好 i 条边后，剩下的图可被若干简单环覆盖（每条边恰好一次）的方案数。对大质数取模。
- $1 \leq n \leq 25$。

IOI2023 集训队互测 R8T1 环覆盖

- 等价于要求每个点的度数都是偶数。所以直接将一条边 (u,v) 看成集合 $\{u,v\}$，那么就是要选出所有集合的对称差为 $\varnothing$。
- 接下来介绍两种做法。这两种做法本质上非常类似，只是提供两种思考方向。

IOI2023 集训队互测 R8T1 环覆盖

- 直接对这 m 个集合计数开桶（称为 $\{c_i\}$）。假如求出了其对称差卷积的 i 次方，那么位置 $\varnothing$ 处的值，就相当于每次选择一个集合，一共选 i 次，最后对称差为 $\varnothing$ 的方案数。顺序有关且不要求互不相同。

IOI2023 集训队互测 R8T1 环覆盖

- 直接对这 m 个集合计数开桶（称为 $\{c_i\}$）。假如求出了其对称差卷积的 i 次方，那么位置 $\varnothing$ 处的值，就相当于每次选择一个集合，一共选 i 次，最后对称差为 $\varnothing$ 的方案数。顺序有关且不要求互不相同。
- 称这个方案数为 g_i，而题目实际要求的方案数为 f_i。发现可以通过 f_i 计算 g_i，通过枚举实际上选了多少个不同的集合，然后分配选择顺序。于是可以利用这个式子 $O(m^2)$ 或更快倒推 f_i。

IOI2023 集训队互测 R8T1 环覆盖

- 直接对这 m 个集合计数开桶（称为 $\{c_i\}$）。假如求出了其对称差卷积的 i 次方，那么位置 $\varnothing$ 处的值，就相当于每次选择一个集合，一共选 i 次，最后对称差为 $\varnothing$ 的方案数。顺序有关且不要求互不相同。
- 称这个方案数为 g_i，而题目实际要求的方案数为 f_i。发现可以通过 f_i 计算 g_i，通过枚举实际上选了多少个不同的集合，然后分配选择顺序。于是可以利用这个式子 $O(m^2)$ 或更快倒推 f_i。
- 问题变为求 g_i。假如能求出 $\hat{c} = \mathrm{FWT}(c)$，那么 g_i 就是 $\hat{c}$ 每一项的 i 次方和。注意到每一项的绝对值都不超过 m 所以这一部分是可以 $O(m^2)$ 计算的。

IOI2023 集训队互测 R8T1 环覆盖

- 直接对这 m 个集合计数开桶（称为 $\{c_i\}$）。假如求出了其对称差卷积的 i 次方，那么位置 $\varnothing$ 处的值，就相当于每次选择一个集合，一共选 i 次，最后对称差为 $\varnothing$ 的方案数。顺序有关且不要求互不相同。
- 称这个方案数为 g_i，而题目实际要求的方案数为 f_i。发现可以通过 f_i 计算 g_i，通过枚举实际上选了多少个不同的集合，然后分配选择顺序。于是可以利用这个式子 $O(m^2)$ 或更快倒推 f_i。
- 问题变为求 g_i。假如能求出 $\hat{c} = \mathrm{FWT}(c)$，那么 g_i 就是 $\hat{c}$ 每一项的 i 次方和。注意到每一项的绝对值都不超过 m 所以这一部分是可以 $O(m^2)$ 计算的。
- 根据 FWT 定义式，$\hat{c}_S$ 就是完全在 S 里的边数减去一端在 S 里的边数。只需要能快速求一个点到一个点集的边数，那么在 S 加入或删除一个点时，这个变化很好做到 $O(2^n)$ 维护。

IOI2023 集训队互测 R8T1 环覆盖

- 直接对这 m 个集合计数开桶（称为 $\{c_i\}$）。假如求出了其对称差卷积的 i 次方，那么位置 $\varnothing$ 处的值，就相当于每次选择一个集合，一共选 i 次，最后对称差为 $\varnothing$ 的方案数。顺序有关且不要求互不相同。
- 称这个方案数为 g_i，而题目实际要求的方案数为 f_i。发现可以通过 f_i 计算 g_i，通过枚举实际上选了多少个不同的集合，然后分配选择顺序。于是可以利用这个式子 $O(m^2)$ 或更快倒推 f_i。
- 问题变为求 g_i。假如能求出 $\hat{c} = \mathrm{FWT}(c)$，那么 g_i 就是 $\hat{c}$ 每一项的 i 次方和。注意到每一项的绝对值都不超过 m 所以这一部分是可以 $O(m^2)$ 计算的。
- 根据 FWT 定义式，$\hat{c}_S$ 就是完全在 S 里的边数减去一端在 S 里的边数。只需要能快速求一个点到一个点集的边数，那么在 S 加入或删除一个点时，这个变化很好做到 $O(2^n)$ 维护。
- 时间复杂度 $O(2^n + m^2)$。

IOI2023 集训队互测 R8T1 环覆盖

- 把每个集合看成一个带形式幂级数元 y 的集合幂级数 $1 + x^S y$，那么要求的直接就是所有乘起来的 $x^{\varnothing} y^i$ 的系数。

IOI2023 集训队互测 R8T1 环覆盖

- 把每个集合看成一个带形式幂级数元 y 的集合幂级数 $1+x^S y$，那么要求的直接就是所有乘起来的 $x^{\varnothing}y^i$ 的系数。
- 每个集合幂级数各自做 FWT 后，每一项形如 $1+y$ 或 $1-y$，那么乘起来的第 S 项形如 $(1+y)^{a_S}(1-y)^{m-a_S}$。现在要求出所有 a_S。

IOI2023 集训队互测 R8T1 环覆盖

- 把每个集合看成一个带形式幂级数元 y 的集合幂级数 $1+x^Sy$，那么要求的直接就是所有乘起来的 $x^{\varnothing}y^i$ 的系数。
- 每个集合幂级数各自做 FWT 后，每一项形如 $1+y$ 或 $1-y$，那么乘起来的第 S 项形如 $(1+y)^{a_S}(1-y)^{m-a_S}$。现在要求出所有 a_S。
- 这个可以对所有 $1+x^S$ 求和后做 FWT，发现 a_S 就是 FWT 后第 S 项的一半（可以理解成代入 $y=1$ 后求了所有 FWT 的和，那么这一项形如 $1+y$ 的贡献就是 2，形如 $1-y$ 的贡献就是 0）。至于实际做这个 FWT 的过程与做法一的最后是一样的 $O(2^n)$。

IOI2023 集训队互测 R8T1 环覆盖

- 把每个集合看成一个带形式幂级数元 y 的集合幂级数 $1+x^S y$，那么要求的直接就是所有乘起来的 $x^{\varnothing} y^i$ 的系数。
- 每个集合幂级数各自做 FWT 后，每一项形如 $1+y$ 或 $1-y$，那么乘起来的第 S 项形如 $(1+y)^{a_S}(1-y)^{m-a_S}$。现在要求出所有 a_S。
- 这个可以对所有 $1+x^S$ 求和后做 FWT，发现 a_S 就是 FWT 后第 S 项的一半（可以理解成代入 $y=1$ 后求了所有 FWT 的和，那么这一项形如 $1+y$ 的贡献就是 2，形如 $1-y$ 的贡献就是 0）。至于实际做这个 FWT 的过程与做法一的最后是一样的 $O(2^n)$。
- 那么现在要求的是 $(1+y)^{a_S}(1-y)^{m-a_S}$ 的 IFWT 的第 $\varnothing$ 次的关于 y 的每一项，也即这些关于 y 的多项式的和。同样利用 a_S 范围是 $O(m)$ 来做到 $O(m^2)$ 计算。

IOI2023 集训队互测 R8T1 环覆盖

- 把每个集合看成一个带形式幂级数元 y 的集合幂级数 $1+x^S y$，那么要求的直接就是所有乘起来的 $x^{\varnothing}y^i$ 的系数。
- 每个集合幂级数各自做 FWT 后，每一项形如 $1+y$ 或 $1-y$，那么乘起来的第 S 项形如 $(1+y)^{a_S}(1-y)^{m-a_S}$。现在要求出所有 a_S。
- 这个可以对所有 $1+x^S$ 求和后做 FWT，发现 a_S 就是 FWT 后第 S 项的一半（可以理解成代入 $y=1$ 后求了所有 FWT 的和，那么这一项形如 $1+y$ 的贡献就是 2，形如 $1-y$ 的贡献就是 0）。至于实际做这个 FWT 的过程与做法一的最后是一样的 $O(2^n)$。
- 那么现在要求的是 $(1+y)^{a_S}(1-y)^{m-a_S}$ 的 IFWT 的第 $\varnothing$ 次的关于 y 的每一项，也即这些关于 y 的多项式的和。同样利用 a_S 范围是 $O(m)$ 来做到 $O(m^2)$ 计算。
- 时间复杂度 $O(2^n+m^2)$。

- 给定一个 n 个点 m 条边的无向连通图，只有 a, b 两种边权 $(1 \le a < b)$。
- 考虑这张图的所有最小生成树。对于每个点 i，求出 1 到 i 在所有最小生成树上的最短路的最小值。
- $2 \le n \le 70, 1 \le m \le 200$。

CF1149D Abandoning Roads

- 首先考虑一条简单路径有没有可能出现在最小生成树上，也就是试图把上面所有的边都放进生成树里。

CF1149D Abandoning Roads

- 首先考虑一条简单路径有没有可能出现在最小生成树上，也就是试图把上面所有的边都放进生成树里。
- 对于边权为 a 的边，必然是都可以放进去的。对于边权为 b 的边，整张图在只保留 a 边的情况下会形成若干个连通块，若对这些连通块缩点，那么我们想强制加进去的边必须不成环。这是充要条件。

CF1149D Abandoning Roads

- 首先考虑一条简单路径有没有可能出现在最小生成树上，也就是试图把上面所有的边都放进生成树里。
- 对于边权为 a 的边，必然是都可以放进去的。对于边权为 b 的边，整张图在只保留 a 边的情况下会形成若干个连通块，若对这些连通块缩点，那么我们想强制加进去的边必须不成环。这是充要条件。
- 所以一条路径合法当且仅当不会离开一个连通块后再回来。但是连通块数仍然可以达到 n 个，看起来又变成旅行商问题了。

CF1149D Abandoning Roads

- 实际上走出去再回来是个很劣的操作，为什么不能直接在连通块里走了完事呢？事实上，如果特判掉连接同一个连通块的 b 边，那么走出去再回来至少需要 $2b$ 的代价。而在连通块大小 ≤ 3 时，一个连通块内用至多 $2a$ 的代价就可以互达了。

CF1149D Abandoning Roads

- 实际上走出去再回来是个很劣的操作，为什么不能直接在连通块里走了完事呢？事实上，如果特判掉连接同一个连通块的 b 边，那么走出去再回来至少需要 $2b$ 的代价。而在连通块大小 ≤ 3 时，一个连通块内用至多 $2a$ 的代价就可以互达了。
- 所以即使不考虑 ≤ 3 的连通块，也不会影响答案的正确性。而大小 ≥ 4 的连通块至多 $\frac{n}{4}$ 个，直接状压后跑分层图最短路就行了。

CF1149D Abandoning Roads

- 实际上走出去再回来是个很劣的操作，为什么不能直接在连通块里走了完事呢？事实上，如果特判掉连接同一个连通块的 b 边，那么走出去再回来至少需要 $2b$ 的代价。而在连通块大小 ≤ 3 时，一个连通块内用至多 $2a$ 的代价就可以互达了。
- 所以即使不考虑 ≤ 3 的连通块，也不会影响答案的正确性。而大小 ≥ 4 的连通块至多 $\frac{n}{4}$ 个，直接状压后跑分层图最短路就行了。
- 时间复杂度 $O(2^{n/4} m \log n)$。由于只有两种边权也可以做到 $O(2^{n/4} m)$。

CF1149D Abandoning Roads

- 实际上走出去再回来是个很劣的操作，为什么不能直接在连通块里走了完事呢？事实上，如果特判掉连接同一个连通块的 b 边，那么走出去再回来至少需要 $2b$ 的代价。而在连通块大小 ≤ 3 时，一个连通块内用至多 $2a$ 的代价就可以互达了。
- 所以即使不考虑 ≤ 3 的连通块，也不会影响答案的正确性。而大小 ≥ 4 的连通块至多 $\frac{n}{4}$ 个，直接状压后跑分层图最短路就行了。
- 时间复杂度 $O(2^{n/4} m \log n)$。由于只有两种边权也可以做到 $O(2^{n/4} m)$。
- 其实理论复杂度可以做到更优，比如多记录一些走过的前几条 b 边的信息，要考虑的连通块个数可以更少。不过不重要了。

CF1261F Xor-Set

- 给定两个集合 A, B，问集合 $C = \{x | x = a \oplus b, a \in A, b \in B\}$ 的所有元素和。对大质数取模。
- 集合 A 由 na 个区间 $[la_i, ra_i]$ 的并给出，集合 B 由 nb 个区间 $[lb_i, rb_i]$ 的并给出。
- $1 \leq na, nb \leq 100, 1 \leq la_i \leq ra_i \leq 10^{18}, 1 \leq lb_i \leq rb_i \leq 10^{18}$。

CF1261F Xor-Set

- 看到区间和二进制很难不拆啊。直接在 Trie 上把所有区间用类似线段树的方法拆开。考虑 A 的一个长 2^a 的区间和 B 的一个长为 2^b 的区间，不妨设 $a \geq b$，那么左右各选一个异或出来的取值范围也一定是一个长 2^a 的区间，高位就是这两个区间的高位做异或，最后做个区间合并就行了。

CF1261F Xor-Set

- 看到区间和二进制很难不拆啊。直接在 Trie 上把所有区间用类似线段树的方法拆开。考虑 A 的一个长 2^a 的区间和 B 的一个长为 2^b 的区间，不妨设 $a \geq b$，那么左右各选一个异或出来的取值范围也一定是一个长 2^a 的区间，高位就是这两个区间的高位做异或，最后做个区间合并就行了。
- 两边的区间数可以达到 $O(n \log r)$，这样最后的总区间数是 $O((n \log r)^2)$，还要区间并，应该是不能接受的。

CF1261F Xor-Set

- 看到区间和二进制很难不拆啊。直接在 Trie 上把所有区间用类似线段树的方法拆开。考虑 A 的一个长 2^a 的区间和 B 的一个长为 2^b 的区间，不妨设 $a \geq b$，那么左右各选一个异或出来的取值范围也一定是一个长 2^a 的区间，高位就是这两个区间的高位做异或，最后做个区间合并就行了。
- 两边的区间数可以达到 $O(n \log r)$，这样最后的总区间数是 $O((n \log r)^2)$，还要区间并，应该是不能接受的。
- 仍然是考虑左边 2^a 和右边 2^b，实际上只保留右边的 a 位以上得到一个长度为 2^a 的区间，这样异或出来和原来是一样的。而对于一个 $[lb_i, rb_i]$ 拆出的区间中，把长度 $\leq 2^a$ 的区间补齐了后本质不同只有 $O(1)$ 个。

CF1261F Xor-Set

- 看到区间和二进制很难不拆啊。直接在 Trie 上把所有区间用类似线段树的方法拆开。考虑 A 的一个长 2^a 的区间和 B 的一个长为 2^b 的区间，不妨设 $a \geq b$，那么左右各选一个异或出来的取值范围也一定是一个长 2^a 的区间，高位就是这两个区间的高位做异或，最后做个区间合并就行了。
- 两边的区间数可以达到 $O(n \log r)$，这样最后的总区间数是 $O((n \log r)^2)$，还要区间并，应该是不能接受的。
- 仍然是考虑左边 2^a 和右边 2^b，实际上只保留右边的 a 位以上得到一个长度为 2^a 的区间，这样异或出来和原来是一样的。而对于一个 $[lb_i, rb_i]$ 拆出的区间中，把长度 $\leq 2^a$ 的区间补齐了后本质不同只有 $O(1)$ 个。
- 所以每个二进制位上实际只需要考虑 $O(n^2)$ 个区间。总区间数是 $O(n^2 \log r)$，可以接受。

- 对于一棵标准编号的无穷二叉树（也即根节点编号为 1，点 x 的左右儿子编号分别为 $2x$ 和 $2x+1$）。
- 给定正整数 s，问该二叉树上有多少条路径的点编号和为 s。对大质数取模。
- $1 \le s \le 10^{15}$。

CF750G New Year and Binary Tree Paths

- 首先考虑直上直下的路径。我们把这上面的点编号写成二进制后一行行写，最低位对齐，例如：

$$\begin{array}{r} 11010010 \\ 110100101 \\ 1101001011 \\ 11010010110 \end{array}$$

- 时刻牢记每条左下右上的对角线必须是一样的。

CF750G New Year and Binary Tree Paths

- 枚举点数 l。考虑右下角这个直角边长 $l-1$ 的三角形，这一部分贡献最大值在全部填 1 时取到，是 2^l-l-1。而设最浅点的编号是 x，贡献是 $x\times(2^l-1)$。所以实际上 x 的选择至多 1 种，就是 $\lfloor\frac{s}{2^l-1}\rfloor$。那么现在就是要右下角的三角形贡献等于 $s'=s \bmod (2^l-1)$。

CF750G New Year and Binary Tree Paths

- 枚举点数 l。考虑右下角这个直角边长 $l-1$ 的三角形，这一部分贡献最大值在全部填 1 时取到，是 2^l-l-1。而设最浅点的编号是 x，贡献是 $x\times(2^l-1)$。所以实际上 x 的选择至多 1 种，就是 $\lfloor\frac{s}{2^l-1}\rfloor$。那么现在就是要右下角的三角形贡献等于 $s'=s \bmod (2^l-1)$。
- 这个从左到右一列一列求和，设 $f_{i,j,k}$ 表示目前填了 i 及以上的位，前面已经有了 j 个 1，还需要后面往前面进位 k 的方案数。转移就看新加的一位是 0 还是 1。因为后面往前面进位最多是 $O(l)$ 的，所以 k 这一维需记到 $O(l)$，加上外层枚举时间复杂度 $O(\log^4 s)$。

CF750G New Year and Binary Tree Paths

- 再考虑非直上直下的路径。同样枚举左右点数 l, r（包括最浅点 x），注意 $2x, 2x+1$ 都固定了要选，那么与 x 有关的部分贡献是 $x + 2x \times (2^{l-1} - 1) + (2x+1) \times (2^{r-1} - 1) = 2^{r-1} - 1 + (2^l + 2^r - 5)x$，无关的贡献是两个三角形，边长分别为 $l - 2$ 和 $r - 2$。

CF750G New Year and Binary Tree Paths

- 再考虑非直上直下的路径。同样枚举左右点数 l, r（包括最浅点 x），注意 $2x, 2x+1$ 都固定了要选，那么与 x 有关的部分贡献是 $x + 2x \times (2^{l-1} - 1) + (2x+1) \times (2^{r-1} - 1) = 2^{r-1} - 1 + (2^l + 2^r - 5)x$，无关的贡献是两个三角形，边长分别为 $l-2$ 和 $r-2$。
- 那么同样 x 的选择至多一种，现在要算两个三角形和为定值的方案数。发现我们不需要区分两个三角形已经填了的 1 的个数，所以状态仍然只有三维，转移就看两边分别填什么。

CF750G New Year and Binary Tree Paths

- 再考虑非直上直下的路径。同样枚举左右点数 l, r（包括最浅点 x），注意 $2x, 2x+1$ 都固定了要选，那么与 x 有关的部分贡献是 $x + 2x \times (2^{l-1} - 1) + (2x+1) \times (2^{r-1} - 1) = 2^{r-1} - 1 + (2^l + 2^r - 5)x$，无关的贡献是两个三角形，边长分别为 $l-2$ 和 $r-2$。
- 那么同样 x 的选择至多一种，现在要算两个三角形和为定值的方案数。发现我们不需要区分两个三角形已经填了的 1 的个数，所以状态仍然只有三维，转移就看两边分别填什么。
- 时间复杂度 $O(\log^5 s)$。

CF1062F Upgrading Cities

- 给定一个 n 个点 m 条边的简单 DAG。
- 问有多少个点 u，满足至多存在一个点 $v \neq u$，从 u 不可达 v 且从 v 不可达 u。
- $1 \leq n, m \leq 3 \times 10^5$。

CF1062F Upgrading Cities

- 首先考虑如何判断一个点是否所有点都可达。因为是 DAG 所以可以考虑拓扑序，以下为叙述方便直接把点重编号为拓扑序。

CF1062F Upgrading Cities

- 首先考虑如何判断一个点是否所有点都可达。因为是 DAG 所以可以考虑拓扑序，以下为叙述方便直接把点重编号为拓扑序。
- 求解 u 时分为 $< u$ 和 $> u$ 两部分考虑。以 $< u$ 的为例，若一个 $v < u$ 存在一条出边 (v, w) 使得 $w < u$，那么 v 不能到达 u 就蕴含了 w 不能到达 u，那么就没有判断 v 的必要。

CF1062F Upgrading Cities

- 首先考虑如何判断一个点是否所有点都可达。因为是 DAG 所以可以考虑拓扑序，以下为叙述方便直接把点重编号为拓扑序。
- 求解 u 时分为 $< u$ 和 $> u$ 两部分考虑。以 $< u$ 的为例，若一个 $v < u$ 存在一条出边 (v, w) 使得 $w < u$，那么 v 不能到达 u 就蕴含了 w 不能到达 u，那么就没有判断 v 的必要。
- 在只保留有用的 v 后，v 可达 u 就当且仅当 v 有直接连向 u 的边。所以设 r_v 表示 v 点的出边连向的最小值（若不存在则为 $n+1$），那么所有 $< u$ 的点都可达 u，当且仅当所有 $v < u$ 都满足 $r_v \leq u$。

CF1062F Upgrading Cities

- 回到原问题。若存在 $v < u$ 满足 $r_v > u$，那么就在删掉 v 的图上再判断一次（如果存在 ≥ 2 个这样的 v 自然已经不行了）。删掉 v 会影响到以 v 为最小出点的那些点。所以设 R_w 表示次小值，那么对于所有 $r_w = v$ 的 w，需要 $R_w \leq u$。

CF1062F Upgrading Cities

- 回到原问题。若存在 $v < u$ 满足 $r_v > u$，那么就在删掉 v 的图上再判断一次（如果存在 ≥ 2 个这样的 v 自然已经不行了）。删掉 v 会影响到以 v 为最小出点的那些点。所以设 R_w 表示次小值，那么对于所有 $r_w = v$ 的 w，需要 $R_w \leq u$。
- 时间复杂度 $O(n + m)$。

CF1062F Upgrading Cities

- 回到原问题。若存在 $v < u$ 满足 $r_v > u$，那么就在删掉 v 的图上再判断一次（如果存在 ≥ 2 个这样的 v 自然已经不行了）。删掉 v 会影响到以 v 为最小出点的那些点。所以设 R_w 表示次小值，那么对于所有 $r_w = v$ 的 w，需要 $R_w \leq u$。
- 时间复杂度 $O(n + m)$。
- 顺带一提，这题的洛谷题解做法看起来比较有趣，但感觉多多少少有点假，可能是能修好的但我不太清楚。

CF715E Complete the Permutations

- 给定两个 1 到 n 的排列 $\{p_i\},\{q_i\}$，但是有些位置有空缺。
- 定义两个排列 p,q 的距离为，每次选择 p 中两个位置交换，使得 p,q 相同的最小交换次数。
- 对每个 $i \in [0, n-1]$，问使得 p,q 距离恰好为 i 的方案数。对大质数取模。
- $1 \le n \le 250$。

CF715E Complete the Permutations

- 众所周知，两个排列的距离就是连边 $p_i \to q_i$ 后的 $n-$ 环数。

CF715E Complete the Permutations

- 众所周知，两个排列的距离就是连边 $p_i \to q_i$ 后的 $n-$ 环数。
- 那么我们先把能连的边连上。现在成了若干个环和若干条链，填好一对数就相当于连边，要把链连成一些环。

CF715E Complete the Permutations

- 众所周知，两个排列的距离就是连边 $p_i \to q_i$ 后的 $n-$ 环数。
- 那么我们先把能连的边连上。现在成了若干个环和若干条链，填好一对数就相当于连边，要把链连成一些环。
- 直接连还是有点折磨了，比如说可能 x 在 p 里已经出现，y 在 q 里已经出现，那 $x \to y$ 这条边是不合法的；再比如都没有出现，那连 $x \to y$ 的方案数实际上和空位数有关。本质是“位置”这个信息很重要。

CF715E Complete the Permutations

- 众所周知，两个排列的距离就是连边 $p_i \to q_i$ 后的 $n-$ 环数。
- 那么我们先把能连的边连上。现在成了若干个环和若干条链，填好一对数就相当于连边，要把链连成一些环。
- 直接连还是有点折磨了，比如说可能 x 在 p 里已经出现，y 在 q 里已经出现，那 $x \to y$ 这条边是不合法的；再比如都没有出现，那连 $x \to y$ 的方案数实际上和空位数有关。本质是"位置"这个信息很重要。
- 所以我们改一下连边方式，两类各 n 个点，第一类的 p_i 向第二类的 i 连边，第二类的 i 向第一类的 q_i 连边。这样环数是不变的。现在连边就完全是填一个位置了。

CF715E Complete the Permutations

- 众所周知，两个排列的距离就是连边 $p_i \to q_i$ 后的 $n-$ 环数。
- 那么我们先把能连的边连上。现在成了若干个环和若干条链，填好一对数就相当于连边，要把链连成一些环。
- 直接连还是有点折磨了，比如说可能 x 在 p 里已经出现，y 在 q 里已经出现，那 $x \to y$ 这条边是不合法的；再比如都没有出现，那连 $x \to y$ 的方案数实际上和空位数有关。本质是“位置”这个信息很重要。
- 所以我们改一下连边方式，两类各 n 个点，第一类的 p_i 向第二类的 i 连边，第二类的 i 向第一类的 q_i 连边。这样环数是不变的。现在连边就完全是填一个位置了。
- 现在已有的环肯定动不了了。剩下的链按照开头和结尾可以分成 $11, 12, 21, 22$ 四类。注意连边只能在不同类别的点之间。

CF715E Complete the Permutations

- 可以发现 11 和 22 的个数是一样的，而且每个环都形如 11 [12 12 ... 12] 22 [21 21 ... 21] 11 ...。

CF715E Complete the Permutations

- 可以发现 11 和 22 的个数是一样的，而且每个环都形如 11 [12 12 ... 12] 22 [21 21 ... 21] 11 ...。
- 首先先把 11 和 22 配对后形成若干个环，要算一个第一类斯特林数。

CF715E Complete the Permutations

- 可以发现 11 和 22 的个数是一样的，而且每个环都形如 11 [12 12 ... 12] 22 [21 21 ... 21] 11 ...。
- 首先先把 11 和 22 配对后形成若干个环，要算一个第一类斯特林数。
- 然后 12 要么是会塞在一个 11 和 22 之间，要么是若干个只有 12 的环。先枚举属于后者的 12 有多少个，方案数也是斯特林数，而前者的方案数只和 11 和 22 间的空位数有关，是一个常数。

CF715E Complete the Permutations

- 可以发现 11 和 22 的个数是一样的，而且每个环都形如 11 [12 12 ... 12] 22 [21 21 ... 21] 11 ...。
- 首先先把 11 和 22 配对后形成若干个环，要算一个第一类斯特林数。
- 然后 12 要么是会塞在一个 11 和 22 之间，要么是若干个只有 12 的环。先枚举属于后者的 12 有多少个，方案数也是斯特林数，而前者的方案数只和 11 和 22 间的空位数有关，是一个常数。
- 21 也是同理。于是可以求出这三部分各自贡献了 i 个环的方案数 f_i, g_i, h_i，就做完了。

CF715E Complete the Permutations

- 可以发现 11 和 22 的个数是一样的，而且每个环都形如 11 [12 12 ... 12] 22 [21 21 ... 21] 11 ...。
- 首先先把 11 和 22 配对后形成若干个环，要算一个第一类斯特林数。
- 然后 12 要么是会塞在一个 11 和 22 之间，要么是若干个只有 12 的环。先枚举属于后者的 12 有多少个，方案数也是斯特林数，而前者的方案数只和 11 和 22 间的空位数有关，是一个常数。
- 21 也是同理。于是可以求出这三部分各自贡献了 i 个环的方案数 f_i, g_i, h_i，就做完了。
- 时间复杂度 $O(n^2)$。

Thanks!