图论

tichec

2024 年 8 月 11 日

竞赛图定义

竞赛图是通过在无向完全图中为每个边分配方向而获得的有向图

兰道定理

一个长度为 n 的度数序列 d，是合法的度数序列当且仅当：
将 d 从小到大排序后满足
$\forall 1 \leq k \leq n, \sum_{i=1}^{k} d_i \geq \frac{k(k-1)}{2}$。
其中，$k = n$ 时取到等号。

Connected Components?

Problem

给出一个 n 个点，$\dfrac{n(n-1)}{2} - m$ 条边的无向图，问图中有多少连通分量以及每个连通分量有多少点。

输入中给出 m 对点，表示这一对点之间没有边，否则就是有边

输出连通分量的个数以及每个连通分量有多少个点。输出的点数序列必须为单调不降的。

$1 \le n \le 2 \times 10^5, 1 \le m \le \min(2 \times 10^5, \frac{n(n-1)}{2})$

Connected Components?

Solution

发现删除的边数只有 m 条，那么必定存在一个点，它旁边被删去的边数不超过 $\frac{m}{n}$。

Connected Components?

Solution

发现删除的边数只有 m 条，那么必定存在一个点，它旁边被删去的边数不超过 $\frac{m}{n}$。

将所有与它没有连边的点并成一个连通块，我们仅考虑这个连通块与其它点直接的连边。

Connected Components?

Solution

发现删除的边数只有 m 条，那么必定存在一个点，它旁边被删去的边数不超过 $\frac{m}{n}$。

将所有与它没有连边的点并成一个连通块，我们仅考虑这个连通块与其它点直接的连边。

发现其他点的个数不超过 $\frac{m}{n}$，对于这些点，我们花费 $O(n)$ 的时间暴力合并连通块。

复杂度显然是对的

[ARC163D] Sum of SCC

Problem

考虑一张竞赛图 G，其中有 N 个节点，节点编号为 $1, 2, \ldots, N$，且 G 满足：

对于 G 中的所有边 $u \to v$，恰好有 M 条边满足 $u < v$。

设 $f(G)$ 表示图 G 中的强连通分量数量。请你求出所有满足条件的 G 的 $f(G)$ 之和。

答案对 998244353 取模。

$1 \le N \le 30$，$0 \le M \le \frac{N(N-1)}{2}$。

[ARC163D] Sum of SCC

Solution

一个竞赛图的 SCC 个数等于将其点集划分为两个集合 A, B（可为空集）并满足以下限制的方案数 -1：

对于每条满足 $u \in A, v \in b$ 的边 (u, v)，都满足其方向为 $u \to v$。

[ARC163D] Sum of SCC

Solution

一个竞赛图的 SCC 个数等于将其点集划分为两个集合 A, B（可为空集）并满足以下限制的方案数 -1：

对于每条满足 $u \in A, v \in b$ 的边 (u, v)，都满足其方向为 $u \to v$。

证明这个结论也很简单，考虑将这个图缩点，然后它仍然是一个竞赛图，并且是一个链状 DAG。

考虑它的拓扑序 $p_1, p_2, p_3, \ldots, p_k$（因为它是竞赛图，所以拓扑序唯一），并找到一个分界点 $i(0 \le i \le k)$，将 $p_1, p_2, \ldots, p_i$ 所对应的 SCC 划入 A，将 $p_{i+1}, p_{i+2}, \ldots, p_k$ 所对应的 SCC 划入 B。

[ARC163D] Sum of SCC

Solution

一个竞赛图的 SCC 个数等于将其点集划分为两个集合 A, B（可为空集）并满足以下限制的方案数 -1：

对于每条满足 $u \in A, v \in b$ 的边 (u, v)，都满足其方向为 $u \to v$。

证明这个结论也很简单，考虑将这个图缩点，然后它仍然是一个竞赛图，并且是一个链状 DAG。

考虑它的拓扑序 $p_1, p_2, p_3, \ldots, p_k$（因为它是竞赛图，所以拓扑序唯一），并找到一个分界点 $i(0 \le i \le k)$，将 $p_1, p_2, \ldots, p_i$ 所对应的 SCC 划入 A，将 $p_{i+1}, p_{i+2}, \ldots, p_k$ 所对应的 SCC 划入 B。

可以发现一定只有这样划分才是合法的方案，因为把一个 SCC 分到两个集合，一个集合内拓扑序不连续的情况都是不合法的。

而每个 i 就对应了一种合法的方案，且 i 的取值共有 $k+1$ 种，-1 则变为 k，即 SCC 个数。

[ARC163D] Sum of SCC

Solution

现在我们的目标就变为对上述划分方案计数，而这个计数是简单的，直接 dp 即可：

设 $f_{i,j,k}$ 为加入了 $1 \sim i+j$ 号点，且满足 $|A| = i, |B| = j$，有 k 条满足要求的边的方案数。

时间复杂度为 $O(n^3m)$，其中状态数为 $O(n^2m)$，转移为 $O(n)$。

欧拉公式定义

对于连通平面图 G，有 $|V|-|E|+|F|=2$，其中 $|V|,|E|,|F|$ 分别表示 G 的点数、边数、划分平面数。

欧拉公式定义

对于连通平面图 G，有 $|V|-|E|+|F|=2$，其中 $|V|,|E|,|F|$ 分别表示 G 的点数、边数、划分平面数。

进而推知：对于任意平面图 G，$|V|-|E|+|F|=G$ 的连通块数 $+1$。

[APIO2017] 斑斓之地

Problem

在许多天前的中考中，考场的土地是矩形的，它可以被划分成 R 行 C 列的网格状.

语文考试时，伟大的宋蛇从 (s_r, s_c) 出发在考场的土地上移动，宋蛇连续进行了 M 次移动，每次它会向正北 (N)、正南 (S)、正东 (E) 或正西 (W) 方向移动一格，其经过的所有的格子（包括起点和终点）都会变成含苞的梦想。保证在任一时刻，宋蛇都不会离开这片 R 行 C 列的矩形土地。

现在给出宋蛇 M 次移动的方向，你有 Q 个矩形，问每个矩形内有多少个由宋蛇没有经过的格子构成的连通块。

$0 \le M, Q \le 10^5, 1 \le R, C \le 2 \times 10^5$。(点宋蛇有惊喜哦)

[APIO2017] 斑斓之地

Solution

我们在所有的相邻的可用土地（宋蛇没有经过）之间连边。
令可用土地为白点，其它土地为黑点。

[APIO2017] 斑斓之地

Solution

我们在所有的相邻的可用土地（宋蛇没有经过）之间连边。
令可用土地为白点，其它土地为黑点。
考虑平面图欧拉公式，我们要求的就是 $|V| - |E| + |F|$ 的大小。

[APIO2017] 斑斓之地

Solution

我们在所有的相邻的可用土地（宋蛇没有经过）之间连边。

令可用土地为白点，其它土地为黑点。

考虑平面图欧拉公式，我们要求的就是 $|V|-|E|+|F|$ 的大小。

其中，$|V|$ 即给定矩形区域内白点个数，直接用总点数减黑点数，用主席树维护即可。

将 $|E|$ 分为横边和竖边，每部分也是个二维数点。

[APIO2017] 斑斓之地

Solution

我们在所有的相邻的可用土地（宋蛇没有经过）之间连边。

令可用土地为白点，其它土地为黑点。

考虑平面图欧拉公式，我们要求的就是 $|V| - |E| + |F|$ 的大小。

其中，$|V|$ 即给定矩形区域内白点个数，直接用总点数减黑点数，用主席树维护即可。

将 $|E|$ 分为横边和竖边，每部分也是个二维数点。

对于 $|F|$ ，由于黑点构成的连通块整体上是连通的，我们可以将其分为两类处理：

1. 若干 1×1 大小的小正方形，即它被 4 个白点包裹，这是好求的
2. 如果矩形区域将整条蛇包住且蛇没有经过边界，那么 $|F|$ 要加一

然后就做完了

Prüfer 序列

Prüfer 序列可以将一个带标号 n 个节点的树用 $[1, n]$ 中的 $n-2$ 个整数表示。

你也可以将其理解为完全图的生成树与数列直接的双射。

易用 Prüfer 序列说明完全图 K_n 有 n^{n-2} 棵生成树。

对树建立 Prüfer 序列

每次选择一个编号最小的叶结点并删掉它，然后在序列中记录下它连接到的那个结点。

重复 $n-2$ 次后就只剩下两个结点，算法结束。

很容易发现最后剩下的两个节点中，必定包含 n。

同时每个结点在序列中出现的次数是其度数减 1。

用 Prüfer 序列重建树

根据 Prüfer 序列的性质，我们可以得到原树上每个点的度数。

我们依次枚举 Prüfer 序列中的点，每次我们选择一个度数为 1 的最小的结点编号。

将其与当前枚举到的点连接，然后同时减掉两个点的度。

到最后我们剩下两个度数为 1 的点，其中一个是结点 n，将它们建立连接。

Stranger Trees

Problem

给定一张 n（$2 \le n \le 100$）个节点的无向完全图和这个图的一棵生成树。对于 $i = 0, 1, \cdots, n-1$，求出有多少棵这个完全图的生成树，使得这些生成树与给定的生成树恰好有 i 条边重复。

答案对 $10^9 + 7$ 取模。

Stranger Trees

Solution

先考虑设 g_i 表示至少有 i 条边重合的方案数，这等价于有 $n-i$ 个联通块的方案数。

那么对于 f_i 的计算就只需要二项式反演即可。

Stranger Trees

Solution

先考虑设 g_i 表示至少有 i 条边重合的方案数，这等价于有 $n-i$ 个联通块的方案数。

那么对于 f_i 的计算就只需要二项式反演即可。

然后考虑如果我们已经知道了一个联通块划分 S 之后我们怎么做。

可以证明假设当前有 k 个联通块，第 i 个联通块大小为 a_i，$n=\sum a_i$，则有生成树的数量为 $\prod a \times n^{k-2}$。

证明的过程放在 Matrix-Tree 定理那里。

Stranger Trees

Solution

先考虑设 g_i 表示至少有 i 条边重合的方案数，这等价于有 $n-i$ 个联通块的方案数。

那么对于 f_i 的计算就只需要二项式反演即可。

然后考虑如果我们已经知道了一个联通块划分 S 之后我们怎么做。

可以证明假设当前有 k 个联通块，第 i 个联通块大小为 a_i，$n=\sum a_i$，则有生成树的数量为 $\prod a \times n^{k-2}$。

证明的过程放在 Matrix-Tree 定理那里。

如果考虑 $f_{i,j,k}$ 为 i 为根的子树划分了 j 个联通块，当前点所在的联通块大小为 k 的贡献和是可以轻易做到 $O(n^3)$ 的。

Stranger Trees

Solution

先考虑设 g_i 表示至少有 i 条边重合的方案数，这等价于有 $n-i$ 个联通块的方案数。

那么对于 f_i 的计算就只需要二项式反演即可。

然后考虑如果我们已经知道了一个联通块划分 S 之后我们怎么做。

可以证明假设当前有 k 个联通块，第 i 个联通块大小为 a_i，$n=\sum a_i$，则有生成树的数量为 $\prod a \times n^{k-2}$。

证明的过程放在 Matrix-Tree 定理那里。

如果考虑 $f_{i,j,k}$ 为 i 为根的子树划分了 j 个联通块，当前点所在的联通块大小为 k 的贡献和是可以轻易做到 $\mathcal{O}(n^3)$ 的。

考虑 $\prod a$ 的组合意义，我们可以视为将树划分为若干个联通块，每个联通块内放入一个球的方案数，这样通过树上背包的 trick 直接 dp 复杂度就是 $\mathcal{O}(n^2)$ 了，然后 $\mathcal{O}(n^2)$ 的二项式反演就可以了。

二分图定义

节点由两个集合组成，且两个集合内部没有边的图。

二分图定义

节点由两个集合组成，且两个集合内部没有边的图。

positive1 的课件

匈牙利算法的小优化: 每次 dfs 时如果没找到新的匹配就不清空 vis 数组，在 m 较大时优化较为明显。

二分图最大匹配必定存在必经点。

CF1965F Conference

Problem

宋蛇在准备期初考，争取给老师留下好的印象。

每次会连续准备一定天数。宋蛇每天准备一门课，一门课不能准备多次。

现在有 n 门课，宋蛇可以在 l_i 至 r_i 的任何一天准备。

如果有办法在这段时间的每一天宋蛇都可以准备一门课，则称这段时间是合法的，但宋蛇每门课只能准备一次。

从 1 到 n 的每个 k，有多少个连续的 k 天是合法的。

$1 \le n, l_i, r_i \le 2 \times 10^5$

CF1965F Conference

Solution

先考虑给定一个时间段，怎么判断是否合法？

CF1965F Conference

Solution

先考虑给定一个时间段，怎么判断是否合法？

可以发现是个二分图匹配问题，有个经典的贪心，从左到右扫描线，扫到 i 时，把以 i 为左端点的区间加入堆中，然后找到可以匹配 i 的右端点最小的区间匹配。

如果每个 i 都匹配上了，就合法。

CF1965F Conference

Solution

我们先对全局做一次上述贪心，求出二分图的最大匹配情况。

CF1965F Conference

Solution

我们先对全局做一次上述贪心，求出二分图的最大匹配情况。

考虑 n 个区间中匹配上的那些区间 $a_1, a_2, \cdots a_k$，我们可以发现，任意一个时间段，都存在一组最大匹配，使得匹配上的区间都是 a 中的一个。

证明考虑上述贪心过程，加入时间段限制只会使堆中的元素变多，因此原先没用的区间依旧没用。

CF1965F Conference

Solution

同时我们再对 a 中的区间进行一些处理，如果区间 $[l, r]$ 匹配的是时刻 t，那么我们可以把区间 $[l, r]$ 变为 $[t, r]$，证明是类似的，同样考虑上述贪心过程。

那么我们保证了 a 中的区间两两左端点不同。

CF1965F Conference

Solution

同时我们再对 a 中的区间进行一些处理，如果区间 $[l, r]$ 匹配的是时刻 t，那么我们可以把区间 $[l, r]$ 变为 $[t, r]$，证明是类似的，同样考虑上述贪心过程。

那么我们保证了 a 中的区间两两左端点不同。

这保证了，对于时间段 $[L, R]$，包含于 $[L, R]$ 的区间数量不超过 $R - L + 1$，即区间长度。

CF1965F Conference

Solution

同时我们再对 a 中的区间进行一些处理，如果区间 $[l,r]$ 匹配的是时刻 t，那么我们可以把区间 $[l,r]$ 变为 $[t,r]$，证明是类似的，同样考虑上述贪心过程。

那么我们保证了 a 中的区间两两左端点不同。

这保证了，对于时间段 $[L,R]$，包含于 $[L,R]$ 的区间数量不超过 $R-L+1$，即区间长度。

考虑 Hall 定理，我们要选择一些时间点使得，有交的区间数量减时间点数量最小，最优情况下会选择一段连续的时间。

因为假如不连续，考虑时间段 $[L,R]$ 使得 $L-1$ 和 $R+1$ 选了，但 $[L,R]$ 没选，那么加入 $[L,R]$ 后新增的有交区间为包含于 $[L,R]$ 的区间，数量不超过 $[L,R]$ 的长度。

CF1965F Conference

Solution

那现在只要对于每个左端点 l 求出最小右端点 r 使时间段 $[l,r]$ 不合法，这是容易的，因为区间左端点互不相同，时间段右端点 $+1$ 最多增加 1 个有交区间。时间复杂度 $O(n\log n)$。

CF1408H Rainbow Triples

Problem

给定长度为 n 的序列 p
找出尽可能多的三元组 (a_i, b_i, c_i) 满足：
- $1 \le a_i < b_i < c_i \le n$
- $p_{a_i} = p_{c_i} = 0$，$p_{b_i} \ne 0$
- p_{b_i} 互不相同。
- 所有的 a_i, b_i, c_i 互不相同。

输出最多可以选出多少个三元组，多组数据。
$\sum n \le 5 \cdot 10^5, 0 \le a_i \le n$。

CF1408H Rainbow Triples

Solution

二分答案 k，容易发现最优的一定选择前 k 个 0，后面 k 个 0 然后匹配。

CF1408H Rainbow Triples

Solution

二分答案 k，容易发现最优的一定选择前 k 个 0，后面 k 个 0 然后匹配。

容易发现匹配一定是从左到右依次匹配，就是说包含不优。

CF1408H Rainbow Triples

Solution

二分答案 k，容易发现最优的一定选择前 k 个 0，后面 k 个 0 然后匹配。

容易发现匹配一定是从左到右依次匹配，就是说包含不优。

考虑构建二分图模型，左侧每个点对应一对括号（也就是 0），右侧一个点对应一种颜色，如果某个括号 x 包含某个颜色 y，那么 x, y 有一条边。要求所有括号有匹配。

CF1408H Rainbow Triples

Solution

二分答案 k，容易发现最优的一定选择前 k 个 0，后面 k 个 0 然后匹配。

容易发现匹配一定是从左到右依次匹配，就是说包含不优。

考虑构建二分图模型，左侧每个点对应一对括号（也就是 0），右侧一个点对应一种颜色，如果某个括号 x 包含某个颜色 y，那么 x, y 有一条边。要求所有括号有匹配。

考虑 Hall 定理：对于左侧的每个点集 S，设 S 的出边构成的点集为 T，要求 $|S| \leq |T|$。

CF1408H Rainbow Triples

Solution

放在原序列上其实是若干个区间并起来求颜色数。
发现所有区间都有交的，那么肯定是选择一段连续区间进行判定。

CF1408H Rainbow Triples

Solution

放在原序列上其实是若干个区间并起来求颜色数。

发现所有区间都有交的，那么肯定是选择一段连续区间进行判定。

假设左侧选择从左往右第 i 个 0，右侧选择从右往左第 j 个 0，那么要求：$s[i,j] \geq k-j-i+2$。

推右端点用数据结构维护 $s[i,j]+i$ 的值可以做到单次 $O(n\log n)$ check，总复杂度 $O(n\log^2 n)$，考虑进一步优化。

CF1408H Rainbow Triples

Solution

放在原序列上其实是若干个区间并起来求颜色数。

发现所有区间都有交的，那么肯定是选择一段连续区间进行判定。

假设左侧选择从左往右第 i 个 0，右侧选择从右往左第 j 个 0，那么要求：$s[i,j] \geq k-j-i+2$。

推右端点用数据结构维护 $s[i,j]+i$ 的值可以做到单次 $O(n\log n)$ check，总复杂度 $O(n\log^2 n)$，考虑进一步优化。

发现可以去掉二分，对于一对 i,j，它们的限制实际上是若 $k\geq \max(i,j)$，则 $k\leq s[i,j]+i+j-2$。

同时每个端点对应的标号（i 或 j）是固定的，所以可以只维护一次，复杂度 $O(n\log n)$。

CF590E Birthday

Problem

给定 n 个仅包含 a, b 的字符串，保证它们两两不同。

你需要去掉尽可能少的字符串，使得剩下的字符串中不存在某一个串是另一个串的子串。

$n \le 750$，$\sum_{i=1}^{n} |s_i| \le 10^7$。

CF590E Birthday

Solution

首先可以建 AC 自动机然后求出所有的子串关系。
显然我们不能暴力跳 fail，这样显然会 T 飞。

CF590E Birthday

Solution

首先可以建 AC 自动机然后求出所有的子串关系。

显然我们不能暴力跳 fail，这样显然会 T 飞。

注意到子串关系是一个偏序关系，因此我们只需要跳到最近的结尾同时路径压缩即可，那么此时可以建出一个 DAG。

题目要求的实际上是这个偏序关系的最长反链，通过 Dilworth 定理可以发现就是 DAG 的最小链覆盖。

CF590E Birthday

Solution

首先可以建 AC 自动机然后求出所有的子串关系。

显然我们不能暴力跳 fail，这样显然会 T 飞。

注意到子串关系是一个偏序关系，因此我们只需要跳到最近的结尾同时路径压缩即可，那么此时可以建出一个 DAG。

题目要求的实际上是这个偏序关系的最长反链，通过 Dilworth 定理可以发现就是 DAG 的最小链覆盖。

最小链覆盖又可以通过 $\mathcal{O}(n^3)$ 的传递闭包转化为最小路径覆盖。

CF590E Birthday

Solution

又 DAG 的最小路径点覆盖包含的路径条数 $= n -$ 其拆点二分图的最大匹配数。

CF590E Birthday

Solution

又 DAG 的最小路径点覆盖包含的路径条数 $= n-$ 其拆点二分图的最大匹配数。

使用匈牙利算法即可在 $\mathcal{O}(n^3)$ 的时间求出答案。接下来的问题是如何构造方案。

CF590E Birthday

Solution

又 DAG 的最小路径点覆盖包含的路径条数 $= n-$ 其拆点二分图的最大匹配数。

使用匈牙利算法即可在 $\mathcal{O}(n^3)$ 的时间求出答案。接下来的问题是如何构造方案。

我们先求出传递闭包后的图的最小路径点覆盖包含的路径集合 $path$:

1. 设在拆点二分图中左部点 x 对应的右部点为 x'，若 x, y 匹配则有 $f_x = y, f_y = x$。
2. 依次考虑左部的每一个非匹配点 x_0。
3. 从 x_0 出发，每次从 x 走到 $f_{x'}$，直至到达一个左部点 y_0，满足 y'_0 是非匹配点。
4. 那么经过的所有点构成一条以 y_0 为起点 x_0 为终点的路径。

CF590E Birthday

Solution

接下来我们要从 $path$ 的每条路径上选出一个点构成原图的最长反链：

1. 将所有的终点 x_0 放到一起构成一个集合 E。

2. 求出从 E 中的所有节点出发，走一条边，到达的所有节点 $next(E)$。

3. 根据传递闭包的性质，若 E 与 $next(E)$ 没有交，那么 E 即为所求。

4. 否则考虑 $E \cap next(E)$ 的所有节点 e，沿着 e 所在的路径反着走，直到一个节点 $e' \notin next(E)$，在 E 中将 e 替换为 e'。

5. 回到第 3 步。

CF590E Birthday

Solution

接下来我们要从 $path$ 的每条路径上选出一个点构成原图的最长反链：

1. 将所有的终点 x_0 放到一起构成一个集合 E。

2. 求出从 E 中的所有节点出发，走一条边，到达的所有节点 $next(E)$。

3. 根据传递闭包的性质，若 E 与 $next(E)$ 没有交，那么 E 即为所求。

4. 否则考虑 $E \cap next(E)$ 的所有节点 e，沿着 e 所在的路径反着走，直到一个节点 $e' \notin next(E)$，在 E 中将 e 替换为 e'。

5. 回到第 3 步。

总时间复杂度 $\mathcal{O}(m+n^3)$。注意本题任何与 AC 自动机相关的递归均会爆栈。

矩阵树定理（无向图）

设 G 是一个有 n 个顶点的无向图。定义度数矩阵 $D(G)$ 为：
$D_{ii}(G) = \deg(i), D_{ij} = 0, i \neq j$。
设 $\#e(i,j)$ 为点 i 与点 j 相连的边数，并定义邻接矩阵 A 为：
$A_{ij}(G) = A_{ji}(G) = \#e(i,j), i \neq j$。

矩阵树定理（无向图）

设 G 是一个有 n 个顶点的无向图。定义度数矩阵 $D(G)$ 为：
$D_{ii}(G) = \deg(i), D_{ij} = 0, i \neq j$。
设 $\#e(i,j)$ 为点 i 与点 j 相连的边数，并定义邻接矩阵 A 为：
$A_{ij}(G) = A_{ji}(G) = \#e(i,j), i \neq j$。
定义 Laplace 矩阵（亦称 Kirchhoff 矩阵）L 为：
$L(G) = D(G) - A(G)$。

图 G 的所有生成树个数为 L 去掉第 i 行、第 i 列后的矩阵的行列式值（$\forall i \in [1,n]$）。

矩阵树定理（有向图）

设 G 是一个有 n 个顶点的有向图。定义出度矩阵 $D^{out}(G)$ 为：
$D^{out}_{ii}(G) = \deg^{\text{out}}(i), D^{out}_{ij} = 0, i \neq j$。
类似地定义入度矩阵 $D^{in}(G)$。

矩阵树定理（有向图）

设 G 是一个有 n 个顶点的有向图。定义出度矩阵 $D^{out}(G)$ 为：$D^{out}_{ii}(G) = \deg^{\text{out}}(i), D^{out}_{ij} = 0, i \neq j$。

类似地定义入度矩阵 $D^{in}(G)$。

设 $\#e(i,j)$ 为点 i 指向点 j 的有向边数，并定义邻接矩阵 A 为：$A_{ij}(G) = \#e(i,j), i \neq j$。

定义出度 Laplace 矩阵 L^{out} 为：$L^{out}(G) = D^{out}(G) - A(G)$。

定义入度 Laplace 矩阵 L^{in} 为：$L^{in}(G) = D^{in}(G) - A(G)$。

矩阵树定理（有向图）

设 G 是一个有 n 个顶点的有向图。定义出度矩阵 $D^{out}(G)$ 为：$D^{out}_{ii}(G)=\deg^{\text{out}}(i), D^{out}_{ij}=0, i\neq j$。

类似地定义入度矩阵 $D^{in}(G)$。

设 $\#e(i,j)$ 为点 i 指向点 j 的有向边数，并定义邻接矩阵 A 为：$A_{ij}(G)=\#e(i,j), i\neq j$。

定义出度 Laplace 矩阵 L^{out} 为：$L^{out}(G)=D^{out}(G)-A(G)$。

定义入度 Laplace 矩阵 L^{in} 为：$L^{in}(G)=D^{in}(G)-A(G)$。

图 G 的以 r 为根的所有根向树形图个数为 L^{out} 去掉第 r 行、第 r 列后矩阵的行列式值。

图 G 的以 r 为根的所有叶向树形图个数为 L^{in} 去掉第 r 行、第 r 列后矩阵的行列式值。

证明之前结论

有 k 个联通块，第 i 个联通块大小为 a_i，$n=\sum a_i$，则生成树的数量为 $\prod a \times n^{k-2}$。

证明之前结论

有 k 个联通块，第 i 个联通块大小为 a_i，$n=\sum a_i$，则生成树的数量为 $\prod a \times n^{k-2}$。

证明：

我们先将 S 中的所有联通块缩起来，考虑构建一张新图，我们连接 $i \to j$ 的无向边，但是连接重边数量为 $a_i \times a_j$ 那么根据矩阵树定理，我们可以得到其 Laplace 矩阵为：

$$\begin{bmatrix} a_1(n-a_1) & \dots & \dots & -a_1 \times a_k \\ -a_2 \times a_1 & a_2(n-a_2) & \dots & a_2 \times a_k \\ \dots & \dots & \dots & \dots \\ -a_{k-1} \times a_1 & \dots & a_{k-1} \times (n-a_{k-1}) & -a_{k-1} \times a_k \\ -a_1 \times a_k & \dots & \dots & a_k(n-a_k) \end{bmatrix}$$

然后我们删除一行一列后的行列式即为答案：

$$\begin{bmatrix} a_1(n-a_1) & \dots & \dots \\ -a_2 \times a_1 & a_2(n-a_2) & \dots \\ \dots & \dots & \dots \\ -a_{k-1} \times a_1 & \dots & a_{k-1} \times (n-a_{k-1}) \end{bmatrix}$$

根据行列式的性质，我们把每行的一个公共系数 a_i 都提取出来后得到：

$$\prod_{i=1}^{n-1} a_i \begin{bmatrix} (n-a_1) & \dots & \dots \\ -a_1 & (n-a_2) & \dots \\ \dots & \dots & \dots \\ -a_1 & \dots & (n-a_{k-1}) \end{bmatrix}$$

然后我们将第 $2 \to n-1$ 列都加到第 1 列上会得到：（这里的 n 指联通块数量）

$$\begin{bmatrix} a_k & \dots & \dots \\ a_k & (n-a_2) & \dots \\ \dots & \dots & \dots \\ a_k & \dots & (n-a_{k-1}) \end{bmatrix}$$

然后我们把每行都减去第一行得到：

$$\begin{bmatrix} a_k & \dots & \dots \\ 0 & n & \dots \\ \dots & \dots & \dots \\ 0 & \dots & n \end{bmatrix}$$

于是其行列式为 $n^{k-2}a_k$，所以答案为：$\left(\prod a_i\right)n^{k-2}$

LGV 引理定义

G 是一个有限的带权有向无环图，路径数量是有限的。

起点 $A = \{a_1, \cdots, a_n\}$，终点 $B = \{b_1, \cdots, b_n\}$。

每条边 e 有权 w_e，对于一个有向路径 P，定义 $\omega(P)$ 为路径上所有边权的积。

对任意顶点 a，b，定义 $e(a, b) = \sum_{P:a \to b} \omega(P)$。

设矩阵

$$M = \begin{pmatrix} e(a_1, b_1) & e(a_1, b_2) & \cdots & e(a_1, b_n) \\ e(a_2, b_1) & e(a_2, b_2) & \cdots & e(a_2, b_n) \\ \vdots & \vdots & \ddots & \vdots \\ e(a_n, b_1) & e(a_n, b_2) & \cdots & e(a_n, b_n) \end{pmatrix}$$

定义

从 A 到 B 的不相交路径 $P=(P_1,P_2,\cdots,P_n)$，P_i 表示从 a_i 到 $b_{\sigma(i)}$ 的一条路径，其中 σ 是一个排列，并且满足对任意 $i\neq j$，P_i 与 P_j 没有公共点。

记 $\sigma(P)$ 表示 P 对应的排列。

引理说明，M 的行列式是所有从 A 到 B 的不相交路径 $P=(P_1,\cdots,P_n)$ 的带符号和。

其中符号指 $\sigma(P)$ 的逆序数的奇偶性：$(-1)^{\text{逆序数}}$，记为 $\operatorname{sign}(\sigma(P))$。

$$\det(M)=\sum_{P:A\to B}\operatorname{sign}(\sigma(P))\prod_{i=1}^{n}\omega(P_i)$$

一个不严谨的证明是，存在相交的路径会通过逆序对被消掉。

[PA2021] Fiolki 2

Problem

给定一张 n 个点的 DAG，有 m 条边，保证点 $1 \sim k$ 没有入度。

对每个 $i \in [0, k]$，求出满足条件的区间 $[l, r] \subseteq (k, n]$ 的数量，使得起点在 $[1, k]$ 且终点在 $[l, r]$ 的极大不相交路径组大小为 i。

$n \leq 10^5, m \leq 10^6, k \leq \min(n-1, 50)$

[PA2021] Fiolki 2

Solution

考虑 LGV 引理，发现权值中 $(-1)^{\sigma(p)}$ 很烦，但是本题是判定【是否存在】。

[PA2021] Fiolki 2

Solution

考虑 LGV 引理，发现权值中 $(-1)^{\sigma(p)}$ 很烦，但是本题是判定【是否存在】。

所以可以考虑选定一个大质数 P，然后对每条边赋一个 $[0, P-1]$ 的随机边权。

重新定义一条路径的权值是 $\prod w_i$。

[PA2021] Fiolki 2

Solution

考虑 LGV 引理，发现权值中 $(-1)^{\sigma(p)}$ 很烦，但是本题是判定【是否存在】。

所以可以考虑选定一个大质数 P，然后对每条边赋一个 $[0, P-1]$ 的随机边权。

重新定义一条路径的权值是 $\prod w_i$。

无解时，求出来的行列式值一定 $=0$，有解极大概率 $\neq 0$。

首先预处理出 $e_{u,v}$ 表示起点为 u，终点为 v 的所有路径的权值和。

[PA2021] Fiolki 2

Solution

对于一个区间 $[l,r]$，考虑 $f(l,r)=k$ 是否成立。

把前面写出来的 $e_{i,v}$ 写成，一个 k 维向量为 $V(i)=\{e_{1,i},e_{2,i},\cdots e_{k,i}\}$，那么 $f(l,r)=k$ 就要求 $V(l)\sim V(r)$ 组成的线性空间是否满秩。

[PA2021] Fiolki 2

Solution

对于一个区间 $[l, r]$，考虑 $f(l, r) = k$ 是否成立。

把前面写出来的 $e_{i,v}$ 写成，一个 k 维向量为 $V(i) = \{e_{1,i}, e_{2,i}, \cdots e_{k,i}\}$，那么 $f(l, r) = k$ 就要求 $V(l) \sim V(r)$ 组成的线性空间是否满秩。

若 $f(l, r) = x$ 就相当于可以选出来 x 个起点，x 个终点，拉出来这些可以搞出来一个行列式 $\neq 0$ 的 $x \times x$ 矩阵，也就是说考虑 $V(l) \sim V(r)$ 组成的极大线性无关组大小为 x，也就是线性空间的维数。

[PA2021] Fiolki 2

Solution

直接插线性基复杂度还是带 n^2 的，考虑使用时间戳线性基，用来维护区间线性基。

在记录基底的基础上，维护加入的时间戳，每次更新的时候也需要维护每个基底对应的时间戳，越晚越好。(可以去看 UOJ91 这道题)

[PA2021] Fiolki 2

Solution

直接插线性基复杂度还是带 n^2 的，考虑使用时间戳线性基，用来维护区间线性基。

在记录基底的基础上，维护加入的时间戳，每次更新的时候也需要维护每个基底对应的时间戳，越晚越好。(可以去看 UOJ91 这道题)

注意到 $f(l,r)$ 对于一组定的 l 单调，对于询问，就是计数 $f(l,r)=x$，注意到答案值域仅为 k，求出所有分界点即可。

对 r 进行扫描线，更新线性基。每次拉出来所有时间戳排序然后相邻两两做差即可。时间复杂度 $O(nk^2+mk)$。

广义串并联图定义

不存在同胚于 K_4 的子图的图被称为广义串并联图。
即可以通过删一度点，缩二度点，叠合重边变为一个单点的图。
它有如下性质：
1. 去掉重边后 $m \leq 2n$。
2. 是个平面图。

SOJ2033 算算

Problem

给你个正 n 边形，一些点对之间有线段连接，保证线段只交于顶点处。

宋蛇想要知道对于 $\forall i \in [0, n]$，这个图大小为 i 的独立集数量。

$1 \leq n \leq 5 \times 10^3, 1 \leq m \leq 2 \times n - 3$。

题解

Hoof and Brain P

Problem

给定一个包含 N 个结点和 M 条边的有向图（$2 \le N \le 10^5$, $1 \le M \le 2 \cdot 10^5$），宋蛇们喜欢玩以下的双人游戏。

在图中的不同结点上放置两个指示物（可以用一些与宋蛇相关的物品代替指示物）。每一回合，一名玩家，蛇，选择一个需要沿某一条出边移动的指示物。另一名玩家，珂，选择沿着哪条出边移动该指示物。两个指示物在任何时刻不允许处于同一个结点上。如果在某些时刻珂不能做出合法的行动，则蛇获胜。如果游戏可以无限进行下去，则珂获胜。

给定 Q 个询问（$1 \le Q \le 10^5$），包含两个指示物所在的初始结点。对于每个询问，输出哪名玩家获胜。

Hoof and Brain P

Solution

首先可以注意到，如果某个棋子所在的点没有出边，那么甲只需要选择这个棋子就能够获胜。

我们对反图进行拓扑排序，将能遍历到的点打上标记，并从原图中删除。

这样剩下的所有点都能到环。

Hoof and Brain P

Solution

首先可以注意到，如果某个棋子所在的点没有出边，那么甲只需要选择这个棋子就能够获胜。

我们对反图进行拓扑排序，将能遍历到的点打上标记，并从原图中删除。

这样剩下的所有点都能到环。

现在我们考虑一个只有 1 条出边的节点 x，假设这条边指向的节点是 y。

那么对于任意 x 上的棋子，移动后都一定会到达 y。

Hoof and Brain P

Solution

首先可以注意到，如果某个棋子所在的点没有出边，那么甲只需要选择这个棋子就能够获胜。

我们对反图进行拓扑排序，将能遍历到的点打上标记，并从原图中删除。

这样剩下的所有点都能到环。

现在我们考虑一个只有 1 条出边的节点 x，假设这条边指向的节点是 y。

那么对于任意 x 上的棋子，移动后都一定会到达 y。

如果甲能够通过让两个棋子都移动到 x 获胜，那么甲显然也能够通过让两个棋子都移动到 y 获胜。

Hoof and Brain P

Solution

于是我们可以将 x 和 y 合并，具体来说，我们对于每条边 $z \to x$，将其删除，然后若 $z \to y$ 不存在则加入边 $z \to y$。

Hoof and Brain P

Solution

于是我们可以将 x 和 y 合并，具体来说，我们对于每条边 $z \to x$，将其删除，然后若 $z \to y$ 不存在则加入边 $z \to y$。

和之前相同，此时可能又会有新的节点出边数量变为 1，我们需要将它们继续合并。

合并后，新图中的所有点都有至少 2 条出边，或有一个自环。

Hoof and Brain P

Solution

于是我们可以将 x 和 y 合并，具体来说，我们对于每条边 $z \to x$，将其删除，然后若 $z \to y$ 不存在则加入边 $z \to y$。

和之前相同，此时可能又会有新的节点出边数量变为 1，我们需要将它们继续合并。

合并后，新图中的所有点都有至少 2 条出边，或有一个自环。

先考虑所有点都至少有 2 条出边的情况，显然乙的每次操作都能够避免使两个棋子进入同一个节点。

而有自环的情况同样平凡。

Hoof and Brain P

Solution

于是，对于询问 (x, y)，我们可以按照如下方式判定胜负：若 x 或 y 被删除，那么甲必胜。否则，如果 x 和 y 被合并成了一个节点，那么甲必胜，否则乙必胜。使用启发式合并 set 维护每个节点的出边集合 f 和入边集合 g，并查集维护合并后每个点所在的集合，时间复杂度 $\mathcal{O}(m \log n + q)$。

P8426

Problem

现在有 N 个地点，由 M 条双向道路连接，保证图连通。

宋蛇的家在地点 1，他要去地点 N 上学，他上学一般走最短路，长度为 L。

为了中考和人生目标，宋蛇决定绕路回家。也就是说，他会选择一条从城市 N 到城市 1 且长度大于 L 的路径。

因为宋蛇很困，他不想经过同一座城市多于一次。因此，当他绕远路回家时，不允许经过同一座城市多于一次，并且不允许走回头路。

宋蛇想知道是否存在合法的回家路径。

$2 \leq N \leq 10^5, 1 \leq M \leq 2 \times 10^5$

P8426

Solution

我们假设 $1, n$ 处在同一个点双连通分量，如果不在，则加入边 $(1, n, L)$。

P8426

Solution

我们假设 $1, n$ 处在同一个点双连通分量，如果不在，则加入边 $(1, n, L)$。

发现如果点 u 和 $1, n$ 不在同一个点双连通分量里，那么路径必定不能经过它，将其删除。

这样，原图就是一个点双连通分量了。

P8426

Solution

发现假如这张图同胚于 K_4，即有四个点之间能找到六条边不交的路径将它们两两连接，那么这个图一定无解。

P8426

Solution

发现假如这张图同胚于 K_4，即有四个点之间能找到六条边不交的路径将它们两两连接，那么这个图一定无解。

证明：假设这四个点是 a, b, c, d。

由于 $w(a \to b \to c) = w(a \to d \to b \to c)$，因此 $w(a, d) + w(d, b) = w(a, b)$。

由于 $w(a \to d \to c) = w(a \to b \to d \to c)$，因此 $w(a, d) = w(a, b) + w(b, d)$

P8426

Solution

发现假如这张图同胚于 K_4，即有四个点之间能找到六条边不交的路径将它们两两连接，那么这个图一定无解。

证明：假设这四个点是 a, b, c, d。

由于 $w(a \to b \to c) = w(a \to d \to b \to c)$，因此 $w(a, d) + w(d, b) = w(a, b)$。

由于 $w(a \to d \to c) = w(a \to b \to d \to c)$，因此 $w(a, d) = w(a, b) + w(b, d)$

因此 $w(b, d) = w(a, b) - w(a, d) = w(a, d) - w(a, b) = 0$，又因权值都是正整数，因此假设不成立，这样的图必然有解。

因此我们得到：同胚于 K_4 的点双联通图必定有解。

P8426

Solution

注意到不同胚于 K_4 的点就是广义串并联图，考虑用广义串并联图方法简化这张图：

P8426

Solution

注意到不同胚于 K_4 的点就是广义串并联图，考虑用广义串并联图方法简化这张图：

删一度点：直接删除，不影响答案。

叠合重边：如果两条边权一样无事发生，否则产生一条边权为 $-\infty$ 的边。

缩二度点：产生一条权值为两边权之和的新边。

P8426

Solution

注意到不同胚于 K_4 的点就是广义串并联图，考虑用广义串并联图方法简化这张图：

删一度点：直接删除，不影响答案。

叠合重边：如果两条边权一样无事发生，否则产生一条边权为 $-\infty$ 的边。

缩二度点：产生一条权值为两边权之和的新边。

注意增广的过程中要保证 $1, n$ 两个点不能入队，最后假如的图里只剩一条 $1 \to n$ 的边，且边权不为 $-\infty$ 则无解，否则一定有解。

一般图 Tutte 矩阵

常用于求数据范围较小的一般图最大匹配（很少用来构造方案）。

一般图 Tutte 矩阵

常用于求数据范围较小的一般图最大匹配（很少用来构造方案）。
对于每条边有个变量，设为 $x_{u,v}$。Tutte 矩阵是 $n \times n$ 的。

一般图 Tutte 矩阵

常用于求数据范围较小的一般图最大匹配（很少用来构造方案）。
对于每条边有个变量，设为 $x_{u,v}$。Tutte 矩阵是 $n \times n$ 的。
对于位置 (i, j)，$i < j$，如果存在边 (i, j)，则值为 $x_{i,j}$。
对于位置 (j, i)，$i < j$，如果存在边 (i, j)，则值为 $-x_{i,j}$。
否则为 0。

一般图 Tutte 矩阵

常用于求数据范围较小的一般图最大匹配（很少用来构造方案）。
对于每条边有个变量，设为 $x_{u,v}$。Tutte 矩阵是 $n \times n$ 的。
对于位置 (i,j)，$i<j$，如果存在边 (i,j)，则值为 $x_{i,j}$。
对于位置 (j,i)，$i<j$，如果存在边 (i,j)，则值为 $-x_{i,j}$。
否则为 0。
这个矩阵的秩是偶数，且最大匹配即为秩的一半。
变量不好处理，可以每个变量随机权值，可以证明错误率 $<\frac{n}{p}$，其中 n 为点数，p 为大质数，取 10^9+7 即可。

Tutte 矩阵

该算法一般用于 n 较小的情况。

如果只求最大匹配的数目，则可以用一次高斯消元求出矩阵的秩，比带花树简洁。

Tutte 矩阵

该算法一般用于 n 较小的情况。

如果只求最大匹配的数目，则可以用一次高斯消元求出矩阵的秩，比带花树简洁。

森林邻接矩阵的秩是森林最大匹配的两倍，证明可以看 CF1067E。

P9792 Bimatching

Problem

二分图匹配，但是一个左部点要恰好匹配两个右部点，求最大匹配。
$\sum(n+m) \le 150$。

P9792 Bimatching

Solution

将左部点拆成 x, x'，对于一条边 (x, y)，(x, y) 和 (x', y) 都连边，并连接 (x, x')。

答案即为一般图最大匹配 $-n$。

用上述算法计算即可。

随机化维护图联通 (P10778)

Problem

给定无向图 $G=(V,E)$，q 次询问每次给定一个边集，求删除该边集后图是否连通。保证边集大小不超过 15。强制在线。

$1\le n\le 10^5$，$1\le m\le 5\times 10^5$，$1\le q\le 5\times 10^4$。保证图中没有重边与自环。

随机化维护图联通 (P10778)

Solution

直接做是动态图连通性，不是很好做的。
我们在图上随机找出一棵生成树，对于非树边我们随机一个权值。
对于树边我们令它的权值为经过它的非树边的权值的异或和。

随机化维护图联通 (P10778)

Solution

直接做是动态图连通性，不是很好做的。

我们在图上随机找出一棵生成树，对于非树边我们随机一个权值。

对于树边我们令它的权值为经过它的非树边的权值的异或和。

那么如果输入进的边的权值集合存在一个子集使得异或和为 0。

就可以说明有一条树边和经过它的所有非树边都被断开，即图不连通。

随机化维护图联通 (P10778)

Solution

直接做是动态图连通性，不是很好做的。

我们在图上随机找出一棵生成树，对于非树边我们随机一个权值。

对于树边我们令它的权值为经过它的非树边的权值的异或和。

那么如果输入进的边的权值集合存在一个子集使得异或和为 0。

就可以说明有一条树边和经过它的所有非树边都被断开，即图不连通。

判断是否有子集异或和为 0 是线性基的基本应用

CF1827E Bus Routes

Problem

给定一棵 n 个节点的树。再给定 m 条路径，每条路径 (u, v) 表示 u, v 两点间简单路径上的点可以互相到达。

现在问对于任意两个城市，是否能通过不超过两条路径到达。

如果可以，输出 YES，否则输出 NO，并输出不能通过不超过两条路径到达的两个城市。

$n, m \le 5 \times 10^5$

CF1827E Bus Routes

Solution

随便定一个根 r。考虑一个叶子 l，令 $anc(l)$ 表示 l 通过一条链能到达的最浅祖先。

如果存在两个 $anc(l_1), anc(l_2)$ 不是祖先后代关系，那么 l_1, l_2 就不能通过两条链连通，答案是 NO。

CF1827E Bus Routes

Solution

随便定一个根 r。考虑一个叶子 l，令 $anc(l)$ 表示 l 通过一条链能到达的最浅祖先。

如果存在两个 $anc(l_1), anc(l_2)$ 不是祖先后代关系，那么 l_1, l_2 就不能通过两条链连通，答案是 NO。

否则，我们只需要保留一个最深的 $anc(l)$，设为 v。

如果存在一个点 u，一步到不了 v，那么答案就是 NO，否则答案就是 YES。

有一个偷懒的办法就是直接以 v 为根再求一次 anc，所有点的 anc 都是 v 的话就是 YES。

CF1583H Omkar and Tours

Problem

给定一棵有 n 个节点的树，每个点有点权 e_i，每条边有重量限制 c_i 以及费用 t_i。两点间的费用定义为两点间简单路径上 t_i 的最大值。特别地，若起点与终点相同，则费用为 0。

现在给出 q 个询问，每次给出 v, x，查询从 x 节点出发，只经过 $c_i \geq v$ 的边，能到达的最大点权是多少？前往这些点权最大节点之一，可能的最大费用是多少？

$n, q \leq 2 \times 10^5$

CF1583H Omkar and Tours

Solution

第一问的做法比较简单。我们将询问离线，按照限制从小到大排序，随时加入变为合法的边，并查集维护一下即可。

CF1583H Omkar and Tours

Solution

第一问的做法比较简单。我们将询问离线，按照限制从小到大排序，随时加入变为合法的边，并查集维护一下即可。

难点在于第二问。注意到两点间费用为边权的最大值。我们考虑建立原树的 Kruscal 重构树，就将求路径上最大值转化为了求 lca 的点权。

由于 Kruscal 重构树为一个大根堆，因此深度最小的 lca 即为第二问的答案。

CF1583H Omkar and Tours

Solution

第一问的做法比较简单。我们将询问离线，按照限制从小到大排序，随时加入变为合法的边，并查集维护一下即可。

难点在于第二问。注意到两点间费用为边权的最大值。我们考虑建立原树的 Kruscal 重构树，就将求路径上最大值转化为了求 lca 的点权。

由于 Kruscal 重构树为一个大根堆，因此深度最小的 lca 即为第二问的答案。

我们有引理，要使两点之间 lca 的深度最小，应当尽量让两点的 dfs 序差异尽量大。

由此我们可以知道，一个点到一个联通点集之间最小深度的 lca 有两种情况：从这个点到点集中 dfs 序最小的节点，或者到点集中 dfs 序最大的节点。

然后维护一下 Kruscal 重构树的 lca 即可

CF1989F Simultaneous Coloring

Problem

给定 $n \times m$ 的矩阵。你可以进行两种染色：将某一行染成红色，或将某一列染成蓝色。一次执行一个染色没有代价，一次执行多个染色的代价为 k^2，其中 k 是染色数量。当多个染色同时执行时，对于行和列同时被染色的格子，其颜色可以任选，且不同格子的颜色可以不同。

你需要处理 q 次询问。在每次询问前，所有格子是无色的。在最开始，没有对格子颜色的限制。在第 i 次询问，时额外要求第 x_i 行第 y_i 列的格子必须被染成颜色 c_i，并计算满足当前所有限制的染色方案的最小代价。

$1 \le n, m, q \le 2 \times 10^5$。

CF1989F Simultaneous Coloring

Solution

假设某行或某列被染色两次，那么将前一次染色取消不影响最终的染色结果，而代价不会变大。

因此，可以认为每行每列至多被染色一次。

CF1989F Simultaneous Coloring

Solution

假设某行或某列被染色两次，那么将前一次染色取消不影响最终的染色结果，而代价不会变大。

因此，可以认为每行每列至多被染色一次。

对于一条限制，不妨设染成红色，则要求第 y_i 列的染色不晚于第 x_i 行的染色。

考虑建图，将对每行每列的染色抽象为总共 $n+m$ 个点，u 向 v 连边表示 u 对应的染色不晚于 v，得到一张有向图。

CF1989F Simultaneous Coloring

Solution

假设某行或某列被染色两次，那么将前一次染色取消不影响最终的染色结果，而代价不会变大。

因此，可以认为每行每列至多被染色一次。

对于一条限制，不妨设染成红色，则要求第 y_i 列的染色不晚于第 x_i 行的染色。

考虑建图，将对每行每列的染色抽象为总共 $n+m$ 个点，u 向 v 连边表示 u 对应的染色不晚于 v，得到一张有向图。

对每个强连通分量，必须同时执行其内部所有点对应的染色。

每个大小大于 1 的强连通分量产生大小平方的代价，求和即得答案。

CF1989F Simultaneous Coloring

Solution

问题转化为加边并维护强连通分量，允许离线。

类似整体二分的思想，使用当前区间内时间不超过 mid 的所有边计算第 mid 次询问时图的强连通分量分解。

CF1989F Simultaneous Coloring

Solution

问题转化为加边并维护强连通分量，允许离线。

类似整体二分的思想，使用当前区间内时间不超过 mid 的所有边计算第 mid 次询问时图的强连通分量分解。

对于一条边 $u \to v$，若 u, v 强连通，则丢入 $[l, mid]$ 区间内，否则丢入 $[mid+1, r]$ 区间内。

特别地，若时间超过 mid，则直接丢入 $[mid+1, r]$。

将加入 $[mid+1, r]$ 的边 $u->v$ 的编号改为其所属的强连通分量的编号

CF1989F Simultaneous Coloring

Solution

问题转化为加边并维护强连通分量，允许离线。

类似整体二分的思想，使用当前区间内时间不超过 mid 的所有边计算第 mid 次询问时图的强连通分量分解。

对于一条边 $u \to v$，若 u, v 强连通，则丢入 $[l, mid]$ 区间内，否则丢入 $[mid+1, r]$ 区间内。

特别地，若时间超过 mid，则直接丢入 $[mid+1, r]$。

将加入 $[mid+1, r]$ 的边 $u->v$ 的编号改为其所属的强连通分量的编号

查询即用并查集维护缩点即可。

CF1981E Turtle and Intersected Segments

Problem

宋蛇给你 n 条线段和一个序列 $a_1, a_2, \cdots, a_n$，第 i 条线段是 $[l_i, r_i]$。

宋蛇将按如下方式建图：对于任意的 i, j，若 a_i, a_j 相交，则 i, j 之间连一条边权为 $|a_i - a_j|$ 的边。相交的定义为 $\max(l_1, l_2) \le \min(r_1, r_2)$。

宋蛇想知道最小生成树的边权和是多少。

$n \le 5 \times 10^5$

CF1981E Turtle and Intersected Segments

Solution

对于覆盖一个点的三条边，设其权值为 $(a_1 \le a_2 \le a_3)$ ，发现我们只会在 $(1,2)$ 、$(2,3)$ 之间连边。

CF1981E Turtle and Intersected Segments

Solution

对于覆盖一个点的三条边，设其权值为 $(a_1 \le a_2 \le a_3)$，发现我们只会在 $(1,2)$ 、$(2,3)$ 之间连边。

由此，考虑这样的一个扫描线流程：每条线段在 l 处加入，在 r 处删除。

加入一条线段时找到它按 a 排序后的前驱后继连边。

实际上是维护每个时刻存在的所有线段按 a 排序后连成的一条链。

CF1981E Turtle and Intersected Segments

Solution

对于覆盖一个点的三条边，设其权值为 $(a_1 \le a_2 \le a_3)$，发现我们只会在 $(1,2)$、$(2,3)$ 之间连边。

由此，考虑这样的一个扫描线流程：每条线段在 l 处加入，在 r 处删除。

加入一条线段时找到它按 a 排序后的前驱后继连边。

实际上是维护每个时刻存在的所有线段按 a 排序后连成的一条链。

容易发现做完扫描线后边数缩减到了 $O(n)$，那么直接求最小生成树就行了。

时间复杂度 $O(n \log n)$。

CF1648E Air Reform

Problem

一张 n 个点 m 条边的无向带权简单连通图，其上一条路径的权值为其中所有边权的 $\max$。

考虑这张图的补图，补图中一条边的权值为原图中该边两端点间的最短路（路径长度的定义同上）。保证补图亦是连通图。

对于原图中每条边，求出其两端点在补图中的最短路（定义仍同上）。

$n, m \leq 200\,000$

CF1648E Air Reform

Solution

建出原图 G 的最小生成树 T，则 G 上任意两点 u, v 的最短路即 T 上 u, v 之间边权最大值。

考虑 kruskal 求最小生成树的过程，我们按权值从小到大加入 G 的边 (u, v)。

CF1648E Air Reform

Solution

建出原图 G 的最小生成树 T，则 G 上任意两点 u, v 的最短路即 T 上 u, v 之间边权最大值。

考虑 kruskal 求最小生成树的过程，我们按权值从小到大加入 G 的边 (u, v)。

设 u 对应的连通块为 L，v 对应的连通块为 R 。

加入之前的边，L 和 R 在 G' 上已经形成了若干连通块。

CF1648E Air Reform

Solution

建出原图 G 的最小生成树 T，则 G 上任意两点 u, v 的最短路即 T 上 u, v 之间边权最大值。

考虑 kruskal 求最小生成树的过程，我们按权值从小到大加入 G 的边 (u, v)。

设 u 对应的连通块为 L，v 对应的连通块为 R 。

加入之前的边，L 和 R 在 G' 上已经形成了若干连通块。

考虑暴力合并 L、R 对应的连通块。

依次枚举枚举 L 的连通块 now ，枚举其中的每个点，看其是否 R 对应的连通块中的点有边。

CF1648E Air Reform

Solution

建出原图 G 的最小生成树 T，则 G 上任意两点 u, v 的最短路即 T 上 u, v 之间边权最大值。

考虑 kruskal 求最小生成树的过程，我们按权值从小到大加入 G 的边 (u, v)。

设 u 对应的连通块为 L，v 对应的连通块为 R 。

加入之前的边，L 和 R 在 G' 上已经形成了若干连通块。

考虑暴力合并 L、R 对应的连通块。

依次枚举枚举 L 的连通块 now ，枚举其中的每个点，看其是否 R 对应的连通块中的点有边。

如果无边，删除 R 中的连通块。最后将 now 加入 R 中。

最后的结果便是 R 。

CF1648E Air Reform

Solution

使用启发式合并，考虑枚举 x, y 的复杂度，其与直接连边的次数有关。

因为一旦 x, y 之间在 G 上没有直接连边，那么必然有两个连通块被合并。

CF1648E Air Reform

Solution

使用启发式合并，考虑枚举 x, y 的复杂度，其与直接连边的次数有关。

因为一旦 x, y 之间在 G 上没有直接连边，那么必然有两个连通块被合并。

直接连边的 x, y 每次被枚举一定形如：

1. 在此之前 x 属于 L 的连通块被丢进 R 中，然后 y 是接下来 L 枚举的连通块中的点。因为对 $|L|$ 和 $|R|$ 的合并是启发式的，所以一个点作为 L 中的某个点被合并进 R 的次数不超过 $\log n$。

2. 在 LCA 处，一个作为 L 连通块的点，另一个作为 R 连通块的点被枚举到，这部分枚举次数为 m。

CF1648E Air Reform

Solution

使用启发式合并，考虑枚举 x, y 的复杂度，其与直接连边的次数有关。

因为一旦 x, y 之间在 G 上没有直接连边，那么必然有两个连通块被合并。

直接连边的 x, y 每次被枚举一定形如：

1. 在此之前 x 属于 L 的连通块被丢进 R 中，然后 y 是接下来 L 枚举的连通块中的点。因为对 $|L|$ 和 $|R|$ 的合并是启发式的，所以一个点作为 L 中的某个点被合并进 R 的次数不超过 $\log n$。

2. 在 LCA 处，一个作为 L 连通块的点，另一个作为 R 连通块的点被枚举到，这部分枚举次数为 m。

故直接连边次数为 $\mathcal{O}(m \log n)$，用 set 维护的时间复杂度是 $\mathcal{O}(m \log^2 n)$。

CF1616F Tricolor Triangles

Problem

你有一个 n 个点、m 条边的简单无向图，每条边有颜色编号 c_i，要么被染成 $1,2,3$ 中的一种颜色，要么没有被染色（$c_i=-1$）。

现在，你要对未被染色的边染色，使得对于所有三元环，环上的三条边颜色各不相同或全部相同。

需输出方案。无解输出 -1。

$3 \le n \le 64, 0 \le m \le min(256, \frac{n(n-1)}{2})$

CF1616F Tricolor Triangles

Solution

发现一个边权为 $1,2,3$ 的三元环三边全部相同或不相同有一个共同的条件：三边和为 3 的倍数。

注意到这点就可以暴力把所有三元环找出来，然后根据题目要求列出方程，跑模 3 意义下的高斯消元即可。

根据经典结论，三元环个数为 $\mathcal{O}(m\sqrt{m})$ 的，时间复杂度为 $\mathcal{O}(m^3\sqrt{m})$。

但显然跑不满，常数很小，能过。

CF1906I

Problem

给定 n 个点 $n-1$ 条边的有向弱联通图，问至少增加多少条有向边可以使得图的拓扑序唯一，构造方案。可以证明这个问题一定有解。

$n \le 10^5$

CF1906I

Solution

在原图中，对于每一条链，他的拓扑序都是确定的。

我们需要将原图分成数量尽可能少的链，再将这些链的头尾穿起来即可。

问题转化为求最小链覆盖的方案。

CF1906I

Solution

在原图中，对于每一条链，他的拓扑序都是确定的。

我们需要将原图分成数量尽可能少的链，再将这些链的头尾穿起来即可。

问题转化为求最小链覆盖的方案。

将每个点拆成入点和出点，然后跑二分图最大匹配即可。

先把所有被选上的边拿出来，原图被分成若干条链。

缩完链在原图上跑出拓扑序，按拓扑序将链首尾相连即可。

时间复杂度 $O(n\sqrt{n})$。

CF1867F

Problem

给定一棵树 G，构造一棵树 G'，记 x 为 G' 的一颗子树，对于所有的 x，最大化【x 不与 G 中任意一颗子树同构】的 x 的个数。

同构是有根的。$2 \le n \le 10^6$

CF1867F

Solution

当 $n = 2$ 时，只能构造一条长度为 2 的链。

CF1867F

Solution

当 $n=2$ 时，只能构造一条长度为 2 的链。

当 $n>2$ 时，考虑如果我们找到了一个大小为 w 的树，使得给定的树种没有一个子树与它同构。

那么我们直接从 1 连一条链到这棵子树上，那么就可以做到有 $n-w+1$ 个点满足要求。

那么现在要求的就是最小的 w 。

CF1867F

Solution

当 $n=2$ 时，只能构造一条长度为 2 的链。

当 $n>2$ 时，考虑如果我们找到了一个大小为 w 的树，使得给定的树种没有一个子树与它同构。

那么我们直接从 1 连一条链到这棵子树上，那么就可以做到有 $n-w+1$ 个点满足要求。

那么现在要求的就是最小的 w 。

我们直接从小到大枚子树大小 v ，暴力的枚举树的形态，加上点剪枝。

用树哈希判断是否同构即可。

CF1867F

Solution

当 $n=2$ 时，只能构造一条长度为 2 的链。

当 $n>2$ 时，考虑如果我们找到了一个大小为 w 的树，使得给定的树种没有一个子树与它同构。

那么我们直接从 1 连一条链到这棵子树上，那么就可以做到有 $n-w+1$ 个点满足要求。

那么现在要求的就是最小的 w 。

我们直接从小到大枚子树大小 v ，暴力的枚举树的形态，加上点剪枝。

用树哈希判断是否同构即可。

复杂度：$\Theta(n\cdot f(n))$，$f(n)$ 是 n 个节点无标号有根树个数函数的反函数，不会超过 20，当然这个复杂度远远跑不满。

CF1303F Number of Components

Problem

有一个 $n \times m$（$1 \le n, m \le 300$）的矩阵，初始全是 0。我们定义 $a_{i,j}$ 表示矩阵中第 i 行第 j 列的元素。

如果两个格子有相邻边并且格子中的元素相同，我们就说它们是联通的。联通关系可以传递，也就是说整个矩阵被分成了若干个联通块。

你需要处理 q 次修改操作（$1 \le q \le 2 \times 10^6$），第 i 次操作包含三个整数 x_i, y_i, c_i，表示将 a_{x_i,y_i} 替换为 c_i。

（保证 $1 \le x_i \le n, 1 \le y_i \le m, 1 \le c_i \le \max(1000, \lceil \frac{2 \times 10^6}{nm} \rceil)$）

每次修改结束后你需要求出这个矩阵当前有多少连通块。

注意本题有一个额外限制。对于所有的 $i \in [1, q-1]$，满足 $c_i \le c_{i+1}$。

CF1303F Number of Components

一次操作可以将连通块合并起来，也可以将一个连通块拆成多个连通块，考虑将两种情况分开考虑。

CF1303F Number of Components

一次操作可以将连通块合并起来，也可以将一个连通块拆成多个连通块，考虑将两种情况分开考虑。

对于将连通块合并的情况，用并查集维护即可。

对于将连通块拆开的情况，考虑原矩阵 A ，对 $A_{i,j}$ 进行修改得到 B，我们发现：

$A->B$ 对连通块总数产生的贡献等于 $B->A$ 对连通块总数产生的贡献的相反数。

CF1303F Number of Components

一次操作可以将连通块合并起来，也可以将一个连通块拆成多个连通块，考虑将两种情况分开考虑。

对于将连通块合并的情况，用并查集维护即可。

对于将连通块拆开的情况，考虑原矩阵 A ，对 $A_{i,j}$ 进行修改得到 B，我们发现：

$A->B$ 对连通块总数产生的贡献等于 $B->A$ 对连通块总数产生的贡献的相反数。

然后就可以将操作序列倒过来看（可能会更精彩），将分裂操作变为合并操作。

以最终矩阵为起点，倒着进行操作，计算答案时减去当前贡献即可。

CF1303F Number of Components

一次操作可以将连通块合并起来，也可以将一个连通块拆成多个连通块，考虑将两种情况分开考虑。

对于将连通块合并的情况，用并查集维护即可。

对于将连通块拆开的情况，考虑原矩阵 A，对 $A_{i,j}$ 进行修改得到 B，我们发现：

$A->B$ 对连通块总数产生的贡献等于 $B->A$ 对连通块总数产生的贡献的相反数。

然后就可以将操作序列倒过来看（可能会更精彩），将分裂操作变为合并操作。

以最终矩阵为起点，倒着进行操作，计算答案时减去当前贡献即可。

计算答案，以合并连通块为例：

我们发现对于每一种数字的修改操作都是并在一起的。

直接离线下来对于每种数字用并查集维护一下即可。

第 1 届 ICPC 青少年程序设计竞赛 G.Dynamic Graph

Problem

给定一张 n 个点的无向图，刚开始为空。执行 m 次操作 (3 种操作)。

- 1 u v w，在点 u 与点 v 之间加入一条权值为 w 的边。
- 2 id，删除第 id 次操作加入的边。
- 3 u v w，询问点 u 与点 v 之间是否存在一条权值模 F 为 w 的路径 (不一定要为简单路径)。

$1 \le n, m \le 10^5, 1 \le F \le 10^9$。

第 1 届 ICPC 青少年程序设计竞赛 G.Dynamic Graph

考虑使用线段树分治来解决。
问题在于如何判断模 F 为 w 的路径。

第 1 届 ICPC 青少年程序设计竞赛 G.Dynamic Graph

考虑使用线段树分治来解决。

问题在于如何判断模 F 为 w 的路径。

如果询问的 x, y 不在同一个连通块内则必然无解。否则 x, y 之间的路径权值和必定可以表示为 $L + k \cdot G (\mathrm{mod}\ F)$。

L 表示 x, y 之间固定的一条路径权值和，G 表示 x, y 所在连通块的所有环的长度和 F 的最大公约数。

若能证明任意一条路径权值和能表示成固定的一条路径权值和 L 和若干倍环长最大公约数 G，就可以轻松维护。

第 1 届 ICPC 青少年程序设计竞赛 G.Dynamic Graph

考虑两条 x, y 之间不同的路径的对称差中，一定每个点的度数都是偶数，所以可以将其拆分成若干个环。那么就可以通过环来使得任意两种不同的路径相互转化。

所以就只需要维护环长最大公约数以及任意一条路径长度。

第 1 届 ICPC 青少年程序设计竞赛 G.Dynamic Graph

考虑两条 x, y 之间不同的路径的对称差中，一定每个点的度数都是偶数，所以可以将其拆分成若干个环。那么就可以通过环来使得任意两种不同的路径相互转化。

所以就只需要维护环长最大公约数以及任意一条路径长度。

如果加入的边连接的是在同一连通块的 x, y，则环长的最大公约数 G 更新为 $\gcd(G, 2w, w + L(x) + L(y))$。其中 $L(x)$ 表示 x 到并查集中 x 的根的距离。

否则环长的最大公约数 G 更新为 $\gcd(G_x, G_y, 2w)$，其中 G_x 表示 x 所在连通块的环长的最大公约数。

使用按秩合并的并查集以及线段树合并进行维护即可，时间复杂度 $O(n \log n \log m)$。

CF1392G Omkar and Pies

Problem

给定两个长度为 k 的 01 串，分别是起始串和目标串。

依次给出 n 个可选操作，第 j 个操作 a_j, b_j 表示交换起始串的 a_j 位置与 b_j 位置。

你需要选择操作序列中其中连续的一段操作依次按顺序执行，并且选择的操作数量不小于 m 。

你需要让操作完成后的串与目标串对应相同的位置数量尽可能多。

输出最大数量与你选取操作的区间（若有多种方案输出任意一个即可）。

$1 \le m \le n \le 10^6, 2 \le k \le 20$

CF1392G Omkar and Pies

设启始串为 s ，结束串为 t ，考虑怎么实现操作区间 $[l, r]$ 。
将 s , t 看成二进制数，方便操作。

CF1392G Omkar and Pies

设启始串为 s ，结束串为 t ，考虑怎么实现操作区间 $[l, r]$ 。
将 s , t 看成二进制数，方便操作。
我们发现对于 s 和 t 同时进行操作的答案等价于没操作时的答案。
所以操作 $[l, r]$ 就等价于对 s 操作 $[l, n]$ ，对 t 操作 $[r+1, n]$。

CF1392G Omkar and Pies

设启始串为 s ，结束串为 t ，考虑怎么实现操作区间 $[l,r]$ 。
将 s , t 看成二进制数，方便操作。
我们发现对于 s 和 t 同时进行操作的答案等价于没操作时的答案。
所以操作 $[l,r]$ 就等价于对 s 操作 $[l,n]$ ，对 t 操作 $[r+1,n]$。
考虑怎么计算 a 和 b 的相同数位个数。
设 a 中有 x 个 1 ， b 中有 y 个 1，两者共用 z 个 1。
则相同的数位数量时 $(k-x-y+z)+z$ ，发现 x 和 y 都是固定的。
为了使答案最大，我们只需要使 z 最大即可。

CF1392G Omkar and Pies

考虑对于一个公共 1 状态 sta ，我们需要使得 s 的操作位置小于 t 的操作位置。

CF1392G Omkar and Pies

考虑对于一个公共 1 状态 sta ，我们需要使得 s 的操作位置小于 t 的操作位置。

那么我们只需要求出 $f_{0,sta}, f_{1,sta}$ 分别表示最前面的操作位置使得 $sta \in s$，以及最前面的操作位置使得 $sta \in t$。

发现如果 sta 的状态不是最满的，一定不优。

CF1392G Omkar and Pies

考虑对于一个公共 1 状态 sta ，我们需要使得 s 的操作位置小于 t 的操作位置。

那么我们只需要求出 $f_{0,sta}, f_{1,sta}$ 分别表示最前面的操作位置使得 $sta \in s$，以及最前面的操作位置使得 $sta \in t$。

发现如果 sta 的状态不是最满的，一定不优。

所以我们只用对所有满足 $f_{1,sta} - f_{0,sta} \geq M$ 的状态求一下答案即可。

CF1361E James and the Chase

Problem

给定一个有 n 个点 m 条边的有向强连通图。称一个点是好的当且仅当它到其他点都有且只有一条简单路径。如果好的点至少有 20% 则输出所有好的点，否则输出 -1。

单个测试点内有多组数据。

$1 \le T \le 2\times10^3, 1 \le n \le 10^5, 1 \le m \le 2\times10^5, \sum n \le 10^5, \sum m \le 2\times10^5$。

CF1361E James and the Chase

考虑怎么判断一个点 u 。我们发现以 u 为根建出 dfs 树，如果没有横叉边或前向边就是好节点。

现在就有了 $O(n)$ 的判断一个点是否是好点的方法。

CF1361E James and the Chase

考虑怎么判断一个点 u 。我们发现以 u 为根建出 dfs 树，如果没有横叉边或前向边就是好节点。

现在就有了 $O(n)$ 的判断一个点是否是好点的方法。

考虑以随便一个好点为根，判断其子节点是否合法。

对于一个点 v ，很容易发现一定存在至少一条以它子树内的点为起始的返祖边。

但是如果存在两条以它子树内的点的为起点的返祖边，一定不合法，因为 v 到 fa_v 的简单路径数 > 1。

CF1361E James and the Chase

统计每个点子树内有几条返祖边，首先排除数量 > 1 的点。

CF1361E James and the Chase

统计每个点子树内有几条返祖边，首先排除数量 >1 的点。

假设 v 子树内唯一的一条返祖边指向 w ，发现 v 是好点当且仅当 w 是好点。

1. 如果 w 是好点，那么 w 到所有点路径数为 1 ，可得 v 到子树内、外点路径数都为 1 ，v 也是好点。2. 如果 v 是好点，v 到子树外必经过 w ，w 到子树内必经过 v ，w 也是好点。

CF1361E James and the Chase

统计每个点子树内有几条返祖边，首先排除数量 >1 的点。

假设 v 子树内唯一的一条返祖边指向 w，发现 v 是好点当且仅当 w 是好点。

1. 如果 w 是好点，那么 w 到所有点路径数为 1，可得 v 到子树内、外点路径数都为 1，v 也是好点。2. 如果 v 是好点，v 到子树外必经过 w，w 到子树内必经过 v，w 也是好点。

综上上述，点 $v(v \neq 1)$ 是好点到且仅当 v 子树内连向祖先的边唯一，且连向的那个祖先也是好点。

CF1361E James and the Chase

对于连向祖先的边唯一的条件，可以树上差分判断，同时记录一下连向的祖先的编号，并判断其是否是好的即可。

CF1361E James and the Chase

对于连向祖先的边唯一的条件，可以树上差分判断，同时记录一下连向的祖先的编号，并判断其是否是好的即可。

现在需要找到一个好点，以它为根。

发现题目只要求好点数量 $\geq 20\%$ 时输出，所以随机取 100 个点跑暴力判断即可。

CF1361E James and the Chase

对于连向祖先的边唯一的条件，可以树上差分判断，同时记录一下连向的祖先的编号，并判断其是否是好的即可。

现在需要找到一个好点，以它为根。

发现题目只要求好点数量 $\geq 20\%$ 时输出，所以随机取 100 个点跑暴力判断即可。

如果还是找不到的话直接输出 -1 即可，因为判断不出来的概率 $(\frac{4}{5})^{100}$ 趋近于 0。

找到好点后直接按上述过程求解即可。

Codeforces Gym103119B. Boring Problem

Problem

给定 n 个长度为 m 的字符串 $T_1, T_2, \ldots, T_n$，这些字符串都只包含前 k 个小写字母。

对于询问串 S，考虑以下过程：

Step$_1$：如果 S 包含某个 T_j 作为子串，则结束过程。

Step$_2$：否则，在 S 之后以 p_i 的概率添加第 i 个小写字母，然后回到第 1 步。

定义 $f(S;T,p)$ 为过程结束时 S 的期望长度。

给定一个字符串 R，对于每个 $i = 1, 2 \ldots, |R|$，求出 $f(R[1\ldots i];T,p)$ 对 10^9+7 取模后的结果。

$1 \le n \le 100, 1 \le nm, |R| \le 10^4, 1 \le k \le 26$。

Codeforces Gym103119B. Boring Problem

添加字母的过程相当于在 AC 自动机上移动，终止条件即为碰到叶子节点。

Codeforces Gym103119B. Boring Problem

添加字母的过程相当于在 AC 自动机上移动，终止条件即为碰到叶子节点。

设 E_u 为当前在 u 节点，到过程终止还需的期望步数。则有转移

$$E_u = \begin{cases} 1 + \sum p_i \cdot E_{tr_{u,i}} & u \neq leaf \\ 0 & u = leaf \end{cases}$$

Codeforces Gym103119B. Boring Problem

显然转移会成环，因此高斯消元，AC 自动机上一共有 $O(nm)$ 个点，消元时间复杂度 $O(n^3m^3)$，无法通过。

Codeforces Gym103119B. Boring Problem

显然转移会成环，因此高斯消元，AC 自动机上一共有 $O(nm)$ 个点，消元时间复杂度 $O(n^3m^3)$，无法通过。

考虑将未知数数量减少。先不考虑连向 Fail 指针的边，那么对于一个节点的某个子节点，可以用其它子节点来表示。

Codeforces Gym103119B. Boring Problem

显然转移会成环，因此高斯消元，AC 自动机上一共有 $O(nm)$ 个点，消元时间复杂度 $O(n^3m^3)$，无法通过。

考虑将未知数数量减少。先不考虑连向 Fail 指针的边，那么对于一个节点的某个子节点，可以用其它子节点来表示。

具体地，根据 $E_u = 1 + \sum p_i \cdot E_{tr_{u,i}}$ 可以反推 $E_{tr_{u,x}} = \frac{E_u - 1 - \sum[i \neq x] p_i \cdot E_{tr_{u,i}}}{p_x}$。

此时在每个度数为 deg 的分叉处只新设了 deg − 1 个未知元，加上根处的未知元，整个 AC 自动机一共设了 $1 + \sum(\text{deg}_i - 1) = 1 + \sum_{i \in \text{AC}}[i \neq root] - \sum_{i \in \text{AC}}[i \neq leaf] = \sum_{i \in \text{AC}}[i = leaf] \leq n$ 个未知元。因为可能有重复的字符串所以叶子节点 $\leq n$。

于是神奇地发现此时未知元只有 $O(n)$ 个，那么直接高斯消元即可，时间复杂度 $O(n^3 + nmk)$。

loj3400【2020-2021 集训队作业】Storm

Problem

给出一张 n 个点，m 条边的无向图。每个点有一个点权 a_i，每条边有一个边权 b_i。

设这 m 条边组成的集合为 E，对于一个边集 S，定义其“导出点集” $nxt(S) = \{x | \exists e \in S, x$ 是 e 的一个端点 $\}$。

再给出一个整数 k，求 $max_{|S| \le k} \sum_{x \in nxt(S)} a_x - \sum_{e \in S} b_e$。

换言之，就是选出一个不超过 k 条边的边集 S，最大化（$nxt(S)$ 的点权和减去 S 内的边权和）。

$\sum 2^K(N+M) \le 10^6, 1 \le T \le 5, 0 \le a_i, b_i \le 10^8$ 。

loj3400【2020-2021 集训队作业】Storm

如果选出的边中有三条边构成一条链，则删掉中间的那条一定不劣。
如果选出的边中有若干条构成环，则删掉任何一条一定不劣。

loj3400【2020-2021 集训队作业】Storm

如果选出的边中有三条边构成一条链，则删掉中间的那条一定不劣。

如果选出的边中有若干条构成环，则删掉任何一条一定不劣。

推论：最优解选出的边集，一定构成若干棵直径不超过 2 的树，即若干个不相交的菊花图。

推论：最优解选出的边集，一定构成一张二分图。

loj3400【2020-2021 集训队作业】Storm

如果选出的边中有三条边构成一条链，则删掉中间的那条一定不劣。

如果选出的边中有若干条构成环，则删掉任何一条一定不劣。

推论：最优解选出的边集，一定构成若干棵直径不超过 2 的树，即若干个不相交的菊花图。

推论：最优解选出的边集，一定构成一张二分图。

我们对每个点等概率独立随机地染上黑白两种颜色之一，并要求这一染色方案，恰好也是最优解所对应的二分图的黑白染色方案。

loj3400【2020-2021 集训队作业】Storm

下面，我们尝试计算满足上述要求的解中的最优者。建立费用流模型计算答案：

从 S 向白点连一条容量为 1、费用为 $-A$ 的边，和一条容量为 ∞ 、费用为 0 的边。

从 S 向黑点连一条容量为 1、费用为 $-A$ 的边，和一条容量为 ∞ 、费用为 0 的边。

对于原图中的边 (u, v, B) 满足 u 为白色、v 为黑色，连一条从 u 到 v 的边，容量为 1，费用为 B 。

loj3400【2020-2021 集训队作业】Storm

用基于 SPFA 的连续最短路算法求解该费用流模型，则复杂度是 $O\big(K^2(N+M)\big)$ ，证明：

- 首先，显然 SPFA 的运行次数 $\leq K$ 。
- 然后，在一次 SPFA 中，任何一个结点至多入队 $O(K)$ 次。这是因为：
- 任意时刻有流量的边不会超过 $3K$ 条，否则就意味着在原图中选了超过 K 条边。
- 对于任何一条长为 L 的增广路，其中至少有 $\dfrac{L}{2}-2$ 条边是某条有流量的边的反向边，因为正向边都是从图的左侧指向右侧，只有这些反向边才会从右侧指向左侧。
- 综合以上两条，得到任意一条增广路的长度不超过 $6K+4$ ，因此每个点的入队次数 $\leq 6K+5$ 。
- 综上，复杂度是 $O\big(K^2(N+M)\big)$ 。

loj3400【2020-2021 集训队作业】Storm

引理：如果一枚硬币以 p 的概率掷出正面，则连续掷 $\frac{1}{p} \cdot -\log\epsilon$ 次能以 $1-\epsilon$ 的概率掷出至少一次正面。

- 证明：全反面的概率 $(1-p)^{\frac{1}{p}\cdot-\log\epsilon} = \left((1-p)^{\frac{1}{p}}\right)^{-\log\epsilon} \le \left(\frac{1}{e}\right)^{-\log\epsilon} = \epsilon$ 。

注意到我们已经能够以 $O\big(K^2(N+M)\big)$ 的复杂度获得以 2^{-K} 的概率正确的解。那么我们把整个过程重复 $-2^K\log\epsilon$ 次以得到 $1-\epsilon$ 的正确率，则总复杂度 $O\big(2^K K^2(N+M)\cdot-\log\epsilon\big)$ ，可获得满分。

UOJ 462 新年的小黄鸭

Problem

有一颗 n 个节点的有根树，根节点为 1，要在这颗树上跑树链剖分。

定义树上的一个树链剖分为一种将边集划分为重边和轻边的方法使得每个节点和他的儿子之间最多只有一条重边。

在下面的表示中，我们将重边称为 0 边，轻边称为 1 边。

$f(x, y)$ = 从 x 到 y 依次经过的每一条边的类型拼接而成的一个 01 串 $(x \neq y)$。

对于一个长度为 l 的全 1 串，我们设 $f(x) = l$。

对于一个长度为 l 的全 0 串，我们设 $f(x) = \lceil \log_2 l \rceil + 1$。

对于一个 01 串，我们有 $f(x) = \sum f($ 每一个极大连续全 0/1 子串 $)$。

例如:$f('0011101') = (1 + 1) + 3 + (0 + 1) + 1 = 7$。

对于一个轻重边划分方案 T，我们有 $O(T) = \sum_{i=2}^{n} f((1, i))$。求所有方案中 O 的最小值。

UOJ 462 新年的小黄鸭

设 f_u 表示只考虑 u 子树时的代价。

UOJ 462 新年的小黄鸭

设 f_u 表示只考虑 u 子树时的代价。
枚举重链 (u, v)，其中 v 是叶子节点，代价分为二种：
1. 从重链上延伸出的子树的代价，即 $\sum f_u + siz_u$ 。
2. 重链的代价，记链为 $u - v$ 为 $a_0, ..., a_m$，其中 $a_0 = u, a_m = v$ 。
那么其代价为 $siz_{a_1} + \sum_{i \geq 0} siz_{a_i}$ 。

UOJ 462 新年的小黄鸭

设 f_u 表示只考虑 u 子树时的代价。

枚举重链 (u, v)，其中 v 是叶子节点，代价分为二种：

1. 从重链上延伸出的子树的代价，即 $\sum f_u + siz_u$ 。

2. 重链的代价，记链为 $u - v$ 为 $a_0, \ldots, a_m$，其中 $a_0 = u, a_m = v$ 。那么其代价为 $siz_{a_1} + \sum_{i \geq 0} siz_{a_i}$ 。

考虑用线段树合并维护这个过程。

对于第一种代价，转换成区间即可。

对于第二种，对于每个点，预处理出它的 1 和 $2^i + 1$ 级孩子，转换成区间加即可。

CF1844G

Problem

给你一棵节点编号为 $1 \sim n$（$2 \le n \le 10^5$）的无根树，每条边有未知的正整数边权。

现在对于所有的 $1 \le i < n$，给出点 i 到点 $i+1$ 的距离 d_i（$1 \le d_i \le 10^{12}$），请你还原出任意一组合法的边权或输出 -1 报告无解。

CF1844G

Solution

首先将距离转化成深度相关：$dis(i,j) = dep_i + dep_j - 2dep_{lca(i,j)}$。
所以可以列出方程 $d_i = dep_{i-1} + dep_i - 2dep_{lca(i-1,i)} (i \geq 2)$。

CF1844G

Solution

首先将距离转化成深度相关：$dis(i,j) = dep_i + dep_j - 2dep_{lca(i,j)}$。
所以可以列出方程 $d_i = dep_{i-1} + dep_i - 2dep_{lca(i-1,i)}(i \geq 2)$。
正常解是高斯消元。但是这个方程比较特殊。
注意到 $2dep_{lca(i-1,i)}$ 相当于将 $dep_{lca(i-1,i)}$ 向左平移一位。

CF1844G

Solution

这个警示我们可以从右往左解出 dep_i 每一位的值。

对于每一位，现在问题就转化成了

$x_i + x_{i-1} \equiv d'_i + x'_{lca(i-1,i)} \pmod 2$。

CF1844G

Solution

这个警示我们可以从右往左解出 dep_i 每一位的值。

对于每一位，现在问题就转化成了 $x_i + x_{i-1} \equiv d'_i + x'_{lca(i-1,i)} \pmod 2$。

(x' 表示前一位，x 表示当前位，d' 表示 * 去掉之前位 * 的贡献后当前二进制位上的数)。

这个是容易 $O(n)$ 解出的。总复杂度是 $O(n \log V)$。

CF1662C

Problem

给定一张 n 个点，m 条边的无向图，（边的长度为 1）求有多少条长度为 k 的路径，满足

起点和终点相同

不存在相邻两步走同一条边，即不存在 $a \to b \to a$ 的路径

答案对 998244353 取模

数据范围：$3 \le n \le 100$，$1 \le m \le \frac{n(n-1)}{2}$，$1 \le k \le 10^4$

CF1662C

Solution

假设不存在第二条限制，那就是个很唐的 DP。
设 (x, y) 表示 x 时刻，在 y 这个位置。
我们考虑什么情况不行，即从 $(a, t) -> (b, t+1) -> (a, t+2)$。

CF1662C

Solution

假设不存在第二条限制，那就是个很唐的 DP。

设 (x, y) 表示 x 时刻，在 y 这个位置。

我们考虑什么情况不行，即从 $(a, t) -> (b, t+1) -> (a, t+2)$。

考虑用总方案数 – 不合法方案数。设 $f_{i,j}$ 表示 i 时刻，在 j 这个点的方案数。

那么 $f_{i,j} = \sum_{(v,j)\in E} f_{i-1,v} - f_{i-2,j} \times d_j$。

CF1662C

Solution

假设不存在第二条限制，那就是个很唐的 DP。

设 (x, y) 表示 x 时刻，在 y 这个位置。

我们考虑什么情况不行，即从 $(a, t) -> (b, t+1) -> (a, t+2)$。

考虑用总方案数 − 不合法方案数。设 $f_{i,j}$ 表示 i 时刻，在 j 这个点的方案数。

那么 $f_{i,j} = \sum_{(v,j)\in E} f_{i-1,v} - f_{i-2,j} \times d_j$。

其中，d_j 是 j 的度数减 $i \neq 2$，然后用矩阵维护即可。

CF1036G

Problem

给出一张 DAG（$1 \le n, m \le 10^6$，其中 n 为结点数，m 为边数）

称无入边的结点为“源点”；无出边的结点为“汇点”。我们还保证这张 DAG 的源点数量与汇点数量相等，且均不超过 20 个

现在我们对这张 DAG 重复以下操作：

1. 选择任意一对源点与汇点 s, t

2. 添加一条（有向）边 (t, s)；如果仍还有源点与汇点，就再回到操作 1。可以发现该次操作将会导致 s 不再是一个源点，t 不再是一个汇点；并且该次操作还有可能添加一个自环

现在问，无论操作中的具体选择如何，该图在所有操作结束后，是否总是成为一个强联通分量（即任意一对结点间都可以相互到达）

CF1036G

Solution

设 S 为一些源点 s 的集合，设源点/汇结点的总数 C。
设 $f(S)$ 为所有能被 S 中某一个元素到达的汇结点 t 的集合。

CF1036G

Solution

设 S 为一些源点 s 的集合，设源点/汇结点的总数 C。

设 $f(S)$ 为所有能被 S 中某一个元素到达的汇结点 t 的集合。

我们有结论：

1. 若对于所有可能的 S 满足 $|S| \neq C$（注意这和 $|f(S)| \neq C$ 不等价）且 $|S| \neq 0$，有 $|S| \geq |f(S)|$，那么答案是 NO 。

2. 若不满足 1.，那么答案就是 YES 。

CF1036G

Solution

设 S 为一些源点 s 的集合，设源点/汇结点的总数 C。

设 $f(S)$ 为所有能被 S 中某一个元素到达的汇结点 t 的集合。

我们有结论：

1. 若对于所有可能的 S 满足 $|S| \neq C$（注意这和 $|f(S)| \neq C$ 不等价）且 $|S| \neq 0$，有 $|S| \geq |f(S)|$，那么答案是 NO 。

2. 若不满足 1.，那么答案就是 YES 。

先证明 1，由于 $|S| \geq |f(S)|$，只需考虑将 $f(S)$ 均与 S 内汇结点配对，那么 $f(S)$ 内的汇结点就永远无法到达其它不属于 $f(S)$ 的汇结点了。

CF1036G

Solution

接着证明 2。我们考虑用归纳法证明汇点能到达其它所有汇点：初始时，对于任意汇结点 t 它显然可以到达自己

CF1036G

Solution

接着证明 2。我们考虑用归纳法证明汇点能到达其它所有汇点：

初始时，对于任意汇结点 t 它显然可以到达自己

设 T_t 是 t 可以到达的汇结点的集合，设 $h(T_t)$ 表示 T_t 中所有汇点与源点配对后，配对的源点可以到达的汇结点的集合（设 T_t 配对的源点集合为 S_{T_t} 应有 $|S_{T_t}| = |T_t|$，这里即指 $f(S_{T_t})$）。

那么 $h(T_t)$ 也是可以被 t 到达的。

CF1036G

Solution

接着证明 2。我们考虑用归纳法证明汇点能到达其它所有汇点：

初始时，对于任意汇结点 t 它显然可以到达自己

设 T_t 是 t 可以到达的汇结点的集合，设 $h(T_t)$ 表示 T_t 中所有汇点与源点配对后，配对的源点可以到达的汇结点的集合（设 T_t 配对的源点集合为 S_{T_t} 应有 $|S_{T_t}| = |T_t|$，这里即指 $f(S_{T_t})$）。

那么 $h(T_t)$ 也是可以被 t 到达的。

此时要么 $|T_t| = C$，那么整张图就可以被 t 到达；否则就有 $|h(T_t)|$ 至少为 $|T_t| + 1$，这样归纳下去，t 最后一定能到达整张图。

CF1036G

Solution

接着证明 2。我们考虑用归纳法证明汇点能到达其它所有汇点：

初始时，对于任意汇结点 t 它显然可以到达自己

设 T_t 是 t 可以到达的汇结点的集合，设 $h(T_t)$ 表示 T_t 中所有汇点与源点配对后，配对的源点可以到达的汇结点的集合（设 T_t 配对的源点集合为 S_{T_t} 应有 $|S_{T_t}| = |T_t|$，这里即指 $f(S_{T_t})$）。

那么 $h(T_t)$ 也是可以被 t 到达的。

此时要么 $|T_t| = C$，那么整张图就可以被 t 到达；否则就有 $|h(T_t)|$ 至少为 $|T_t| + 1$，这样归纳下去，t 最后一定能到达整张图。

证明每个汇点最后能到达整张图后，显然整张图就成为了一个强联通分量。

于是枚举 S 检查其 $f(S)$ 即可，时间复杂度 $O(2^{20})$

P10064

Problem

给定一棵 n 个点的无根树。我们希望在一些点对之间修建公交线路，满足任意两个点之间只需要至多两条公交线路就能到达。

形式化地说，考虑树上的所有 $\frac{n(n-1)}{2}$ 条两个端点不同的简单路径。对于这些路径的一个子集 S，称它是好的当且仅当：

- 考虑一张新的图 G，对于一对点 u, v，当且仅当存在 S 中的一条路径 P，满足 u 和 v 都在 P 上，我们会在 u, v 之间连上边权为 1 的无向边。

- 要求 G 中任意两点之间的距离都不超过 2。

你需要求出有多少个子集 S 是好的。由于答案可能很大，输出对 998244353 取模的结果。

$1 \le n \le 3000$。

P10064

Solution

显然我们只需要考虑每个叶子节点 u 新图上距离为 1 的点 S_u。
原题即要求每个叶子 u, v，S_u, S_v 的交非空。

P10064

Solution

显然我们只需要考虑每个叶子节点 u 新图上距离为 1 的点 S_u。
原题即要求每个叶子 u, v，S_u, S_v 的交非空。
S_u 显然是一个连通块，连通块两两交非空等价于有公共元素。
这里公共元素必然是一个连通块。

P10064

Solution

显然我们只需要考虑每个叶子节点 u 新图上距离为 1 的点 S_u。

原题即要求每个叶子 u, v，S_u, S_v 的交非空。

S_u 显然是一个连通块，连通块两两交非空等价于有公共元素。

这里公共元素必然是一个连通块。

根据经典的“点减边”容斥，我们只需要算出“连通块包含某个点的方案数和”减去“连通块包含某个边的方案数”即可。

P10064

Solution

考虑怎么算经过 u 的方案数。我们只需要关心叶子。
我们以 u 为根，考虑容斥，钦定若干叶子不能和 u 联通。

P10064

Solution

考虑怎么算经过 u 的方案数。我们只需要关心叶子。

我们以 u 为根，考虑容斥，钦定若干叶子不能和 u 联通。

dp 状态是容易的，假设 f_i 表示当前已经钦定 i 个叶子不能和 u 联通，那么考虑 u 和 v 合并。假设 siz, lef 表示当前的子树大小，子树叶子个数，转移：

$tf_{i+j} \leftarrow f_i \times \binom{lef_v}{j} \times 2^{(siz_u-i)\times(siz_v-j)}$

tf 是转移后的 f 数组。这样子单个 u 复杂度已经是 $\mathcal{O}(n^2)$ 了，总复杂度是 $\mathcal{O}(n^3)$。

P10064

Solution

考虑怎么算经过 u 的方案数。我们只需要关心叶子。

我们以 u 为根，考虑容斥，钦定若干叶子不能和 u 联通。

dp 状态是容易的，假设 f_i 表示当前已经钦定 i 个叶子不能和 u 联通，那么考虑 u 和 v 合并。假设 siz, lef 表示当前的子树大小，子树叶子个数，转移：

$tf_{i+j} \leftarrow f_i \times \binom{lef_v}{j} \times 2^{(siz_u - i)\times(siz_v - j)}$

tf 是转移后的 f 数组。这样子单个 u 复杂度已经是 $\mathcal{O}(n^2)$ 了，总复杂度是 $\mathcal{O}(n^3)$。

考虑只容斥子树内元素, 对于子树外元素钦定必须经过，这样子是可以直接算答案的，由于不需要卷积子树外的元素，复杂度只有 $\mathcal{O}(n^2)$。

CF1895G

Problem

给定一个 01 串，你需要给每个节点涂上红色或是蓝色。

如果你给第 i 个节点涂上红色，那么你将获得 r_i 个金币；若涂上蓝色，将获得 b_i 个金币。

之后，你会移除所有被涂上蓝色的字符，并统计剩下字符串中的逆序对个数，并失去逆序对个数个金币。

求出能够得到金币的最大数目。

$n \leq 4 \times 10^5$

CF1895G

Solution

由于要求最大化花费，用总花费减去最少的失去金币数，先建出的最小割模型。

考虑割一条边表示失去这条边对应的金币数。

CF1895G

Solution

由于要求最大化花费，用总花费减去最少的失去金币数，先建出的最小割模型。

考虑割一条边表示失去这条边对应的金币数。

$s_i=1$：$(s,i,r_i),(i,t,b_i)$。

$s_i=0$：$(s,i,b_i),(i,t,r_i)$。

对于 $(i<j,s_i=1,s_j=0)$，连 $(i,j,1)$。

CF1895G

Solution

由于要求最大化花费，用总花费减去最少的失去金币数，先建出的最小割模型。

考虑割一条边表示失去这条边对应的金币数。

$s_i = 1$：$(s, i, r_i), (i, t, b_i)$。

$s_i = 0$：$(s, i, b_i), (i, t, r_i)$。

对于 $(i < j, s_i = 1, s_j = 0)$，连 $(i, j, 1)$。

观察数据范围，发现需要模拟最大流。

CF1895G

Solution

首先我们对于每个点先把 $\min(r_i, b_i)$ 的基础流量流了。
对于一个 1 的点，多了 $r_i - \min(r_i, b_i)$ 的从源点流出的流量。
对于一个 0 的点，少了 $r_i - \min(r_i, b_i)$ 的流量流向汇点。

CF1895G

Solution

首先我们对于每个点先把 $\min(r_i, b_i)$ 的基础流量流了。

对于一个 1 的点，多了 $r_i - \min(r_i, b_i)$ 的从源点流出的流量。

对于一个 0 的点，少了 $r_i - \min(r_i, b_i)$ 的流量流向汇点。

所以我们每次找到一个 0 的点就往前找 1 的点有多少个，并且把它们流过来。

贪心的，我们要让前面的还剩下大于等于 1 流量的 1 点最多，所以要先减去较大的数。

CF1895G

Solution

首先我们对于每个点先把 $\min(r_i, b_i)$ 的基础流量流了。

对于一个 1 的点，多了 $r_i - \min(r_i, b_i)$ 的从源点流出的流量。

对于一个 0 的点，少了 $r_i - \min(r_i, b_i)$ 的流量流向汇点。

所以我们每次找到一个 0 的点就往前找 1 的点有多少个，并且把它们流过来。

贪心的，我们要让前面的还剩下大于等于 1 流量的 1 点最多，所以要先减去较大的数。

所以我们就要实现插入，删除所有 0 的点和对前 k 大的数减一三个操作，使用平衡树解决。

注意前 k 大减一之后就不一定是前 k 大了，要交换一下。

CF1887E

Problem

Alice 和你玩游戏。有一个 $n \times n$ 的网格，初始时没有颜色。Alice 在游戏开始前依次给其中 $2n$ 个格子分别涂上了第 $1 \sim 2n$ 种颜色，并告诉你每个颜色的位置。

接下来的每次操作，你可以选择一个未涂色的格子，由 Alice 在 $2n$ 种颜色中选择一个涂在该格子上，并告诉你该颜色。

如果在某次操作后方格图上存在四个不同颜色的点，且它们的位置形成一个平行于边线的矩形，则输出它们以获得胜利。

你至多进行 10 次操作，请构造一个获胜方案。交互库自适应，也就是说 Alice 的决策与你的选择有关。

$T \le 200, n \le 1000$。

CF1887E

Solution

我们把网格图的行列分别看作点，把格子 (x, y) 涂成颜色 c，看作行 x 向列 y 连一条边权为 c 的边。

那么这是一个左右各有 n 个点的二分图，我们要找的合法矩形就是长度为 4 且边权互不相同的环。

CF1887E

Solution

我们把网格图的行列分别看作点，把格子 (x, y) 涂成颜色 c，看作行 x 向列 y 连一条边权为 c 的边。

那么这是一个左右各有 n 个点的二分图，我们要找的合法矩形就是长度为 4 且边权互不相同的环。

由于共有 $2n$ 个点，已经连了 $2n$ 条不相同的边，则已经连的边一定至少形成一个长度为偶数的环。

考虑可以通过连边把这个环对半拆开：因为整个环上边的颜色都不相同，所以不论新连的边是什么颜色，左右两个环都至少有一个满足环上边的颜色互不相同。

CF1887E

Solution

我们把网格图的行列分别看作点，把格子 (x, y) 涂成颜色 c，看作行 x 向列 y 连一条边权为 c 的边。

那么这是一个左右各有 n 个点的二分图，我们要找的合法矩形就是长度为 4 且边权互不相同的环。

由于共有 $2n$ 个点，已经连了 $2n$ 条不相同的边，则已经连的边一定至少形成一个长度为偶数的环。

考虑可以通过连边把这个环对半拆开：因为整个环上边的颜色都不相同，所以不论新连的边是什么颜色，左右两个环都至少有一个满足环上边的颜色互不相同。

选择满足该条件的一侧，重复上述操作，最终一定会得到一个大小为 4 的环。由于每次令环长变为原来的一半，操作次数不超过 10。

CF1799H

Problem

你有一棵 n 个节点的树。宋蛇需要进行 k 次以下操作：选择其中一条边，删除它，并选择剩下的两个部分中的其中一个进行删除，输出剩余部分中的节点数。

你将会得到一颗初始的树和一个序列。你需要找到一种使得序列与所给序列相等的操作方案数量。

答案可能较大，请对 998244353 取模。如果边或者剩余部分的选择不同，则将其视为不同的方案。

$n \leq 5000, 1 \leq k \leq min(6, n-1)$

CF1799H

Solution

令 f_i 表示 i 在树上最后一个连通块的方案数，只需算出 f 就能得知答案（$ans = \frac{\sum f_i}{s_k}$）。

f 的计算可以考虑合并子树内外信息，使用换根 dp 计算出每个子树 / 子树补使用了某个操作集合的方案数。

转移可以 $O(3^k)$ 暴力做。复杂度 $O(n3^k)$。

CF1284F

Problem

给定两棵大小为 $n(n \leq 2.5 \times 10^5)$ 的树 T_1, T_2。

若你删去 T_1 上的一条边 i 后，可以选择 T_2 上的一条边 j 连接上来而形成树，则称 (i, j) 是匹配的。

对于树 T_1 的一条边，你需要为其标记一个数字/不标记，你标记的数字不能相同，你标记的数字 x 与此边所构成的二元组 (i, x) 是匹配的时候，你才能以数字 x 标记其。

求你最多能标记多少条边。

CF1284F

Solution

「$T_1 - e_1 + e_2$ 仍是树」的条件等价于 e_2 的两端在 $T_1 - e_1$ 中分属不同连通块。

发现这个形式很像二分图最大匹配，考虑 Hall 定理：

CF1284F

Solution

「$T_1 - e_1 + e_2$ 仍是树」的条件等价于 e_2 的两端在 $T_1 - e_1$ 中分属不同连通块。

发现这个形式很像二分图最大匹配，考虑 Hall 定理：

对于 T_1 的边集 S，我们试图计算割掉 S 内所有边之后，两端分属不同连通块的 e_2 数量。

若我们把割掉 S 后的 T_1 中每个点的颜色定义为其所在连通块，所求即为 $n-1$ 再减去 T_2 中同色边数量。

假设颜色为 c 的点有 $F(c)$ 个，则两端颜色均为 c 的边数不超过 $F(c)-1$，故同色边总数不超过 $\sum(F(c)-1) = n-1-|S|$。

CF1284F

Solution

「$T_1 - e_1 + e_2$ 仍是树」的条件等价于 e_2 的两端在 $T_1 - e_1$ 中分属不同连通块。

发现这个形式很像二分图最大匹配，考虑 Hall 定理：

对于 T_1 的边集 S，我们试图计算割掉 S 内所有边之后，两端分属不同连通块的 e_2 数量。

若我们把割掉 S 后的 T_1 中每个点的颜色定义为其所在连通块，所求即为 $n-1$ 再减去 T_2 中同色边数量。

假设颜色为 c 的点有 $F(c)$ 个，则两端颜色均为 c 的边数不超过 $F(c)-1$，故同色边总数不超过 $\sum(F(c)-1) = n-1-|S|$。

故原图一定存在完美匹配，答案一定为 $n-1$。

CF1284F

Solution

当我们加入了一组匹配 $e_1 = (u_1, v_1)$，$e_2 = (u_2, v_2)$ 时，将这两条边删去。

CF1284F

Solution

当我们加入了一组匹配 $e_1 = (u_1, v_1)$，$e_2 = (u_2, v_2)$ 时，将这两条边删去。

两棵树的边都少了一条，为了维持树的形态，考虑缩点，将 T_1 中 u_1，v_1 两点缩起来。

为了保持编号的对应，也将 T_2 中 u_1，v_1 缩起来。

CF1284F

Solution

当我们加入了一组匹配 $e_1 = (u_1, v_1)$，$e_2 = (u_2, v_2)$ 时，将这两条边删去。

两棵树的边都少了一条，为了维持树的形态，考虑缩点，将 T_1 中 u_1，v_1 两点缩起来。

为了保持编号的对应，也将 T_2 中 u_1，v_1 缩起来。

如果此时 T_2 是一棵树，那么就得到了一个规模更小的子问题，考虑找合法匹配的过程。

我们选定一条 $e_2 = (u_2, v_2)$，可能合法的 e_1 在 T_1 中 $u_2 \to v_2$ 的路径上。

我们需要找到两个相邻的、在 $T_2 - e_2$ 中分属不同连通块的点作为 u_1，v_1，而 u_2，v_2 两点分属不同连通块，故合法的 e_1 一定存在。

CF1284F

Solution

1. 在 T_2 中任选一条 $e_2 = (u_2, v_2)$；
2. 在 T_1 中 $u_2 \to v_2$ 的路径上找到相邻的、在 T_2 中分属 e_2 两侧的点 u_1，v_1；
3. 在 T_1，T_2 中分别把 u_1，v_1 缩成一个点，递归到规模减小 1 的子问题。

CF1284F

Solution

1. 在 T_2 中任选一条 $e_2=(u_2,v_2)$；
2. 在 T_1 中 $u_2\to v_2$ 的路径上找到相邻的、在 T_2 中分属 e_2 两侧的点 u_1，v_1；
3. 在 T_1，T_2 中分别把 u_1，v_1 缩成一个点，递归到规模减小 1 的子问题。

第一步在 T_2 中任选，最简单的选择是叶子的父边，将其缩点等价于删掉叶子

用并查集维护 T_2 的缩点，由于 T_2 中缩的点在 T_1 中是连通块，直接在 T_1 的路径上二分出相邻的合法的点即可。

时间复杂度 $O(n\log n)$，可以通过此题。

CF1056H

Problem

有一张 n 个点的图，给你 m 有向条路径，每条路径中每个点最多出现一次。

如果存在两个点 A、B，他们在两条不同的路径中均出现，且在两条路经中 A 到 B 之间的点依次有一个点不同，则输出"Human"，否则输出"Robot"。

数据范围为 3×10^5。

CF1056H

Solution

我们实际上求的是所有数对 $(A, B)(A < B)$，满足所有 $s_A == s_B$ 时 s_{A+1} 相同。

考虑根号分块，设块长为 B。

对于每一个 $|S| > B$ 的串，我们暴力判断它和其他字符串之间的关系。

如果 $|S| <= B$，我们 n^2 枚举当点对 $i\ j$，由 s_i 向 s_j 连一条边权为 s_{i+1} 的边。

最后统一判断。

CF1019C

Problem

给定一个有向图（无自环），求出一个集合 Q，使得其中的点两两之间没有连边，且集合中的点可以走不超过两步到达其他所有不在集合中的点。输出任意一组解。

$1 \leq n, m \leq 10^6$

CF1019C

Solution

考虑如果原图是个 DAG，那我们按照拓扑序选点，每次删除它所有指向的点。

这样我们就得到了一个满足条件的集合，使得其它点都能在一步内到达。

CF1019C

Solution

考虑如果原图是个 DAG，那我们按照拓扑序选点，每次删除它所有指向的点。

这样我们就得到了一个满足条件的集合，使得其它点都能在一步内到达。

考虑一个很假的贪心，每次随便选择一个点 v，然后删掉所有 v 所有指向的顶点。

这样我们就得到了一个集合，使得其它点都能在一步内到达，但集合内点可能存在边。

CF1019C

Solution

考虑如果原图是个 DAG，那我们按照拓扑序选点，每次删除它所有指向的点。

这样我们就得到了一个满足条件的集合，使得其它点都能在一步内到达。

考虑一个很假的贪心，每次随便选择一个点 v，然后删掉所有 v 所有指向的顶点。

这样我们就得到了一个集合，使得其它点都能在一步内到达，但集合内点可能存在边。

但是，如果仅保留集合中的点对应的导出子图，发现这是个 DAG。

那么我们把两个算法结合一下即可。

CF1427G

Problem

给定一个 $n \times n$ 的棋盘，其边界上所有位置均填好了数字。

你需要在剩余 $(n-2) \times (n-2)$ 个位置上填数，在输入时用 -1 描述的格子无法填数。

对于一种填数方案，定义此填数方案的权值为各个相邻的格子的差的绝对值之和。

你需要最小化权值，输出可以得到的最小权值。

保证 $3 \leq n \leq 200$，$-1 \leq a_{i,j} \leq 10^9$。保证边界上的方格均未“破损”。

CF1427G

Solution

考虑若边界的权值只有 1 和 2，如何染色。
考虑建立这样一个网络流模型：

CF1427G

Solution

考虑若边界的权值只有 1 和 2，如何染色。
考虑建立这样一个网络流模型：

1. 源点 S 向所有颜色 1 的点连一条流量无限的边。
2. 所有颜色 2 的点向汇点 T 连一条流量无限的边。
3. 所有相邻的格子之间连一条流量为 1 的边。

CF1427G

Solution

考虑若边界的权值只有 1 和 2，如何染色。
考虑建立这样一个网络流模型：

1. 源点 S 向所有颜色 1 的点连一条流量无限的边。
2. 所有颜色 2 的点向汇点 T 连一条流量无限的边。
3. 所有相邻的格子之间连一条流量为 1 的边。

然后跑最大流得到图的最小割即可。
与 S 连通即代表其颜色为 1，与 T 连通代表其颜色为 2。

CF1427G

Solution

考虑原问题，发现所有点的颜色只会选择在边界出现过的颜色。

那么假设总共出现过 m 中不同的颜色，从小到大为 $v_1, v_2, v_3, \ldots, v_m$。

CF1427G

Solution

考虑原问题，发现所有点的颜色只会选择在边界出现过的颜色。

那么假设总共出现过 m 中不同的颜色，从小到大为 $v_1, v_2, v_3, \ldots, v_m$。

考虑差分，对每种 $v_{i+1} - v_i$ 分别求解。

在统计 $v_{i+1} - v_i$ 的贡献时，我们把 $\leq v_i$ 的瓷砖当作颜色 1，其余瓷砖当作颜色 2。

CF1427G

Solution

考虑原问题，发现所有点的颜色只会选择在边界出现过的颜色。

那么假设总共出现过 m 中不同的颜色，从小到大为 $v_1, v_2, v_3, \ldots, v_m$。

考虑差分，对每种 $v_{i+1} - v_i$ 分别求解。

在统计 $v_{i+1} - v_i$ 的贡献时，我们把 $\le v_i$ 的瓷砖当作颜色 1，其余瓷砖当作颜色 2。

然后按照上述方式建图即可。

CF1427G

Solution

那么现在的问题相当于多次询问最大流，每次询问之前把一个与汇点相连的边改为与源点相连。

如果直接跑时间复杂度显然爆炸，但其实把删除的边对应流量退流一下就好了。

CF1427G

Solution

那么现在的问题相当于多次询问最大流，每次询问之前把一个与汇点相连的边改为与源点相连。

如果直接跑时间复杂度显然爆炸，但其实把删除的边对应流量退流一下就好了。

考虑证明时间复杂度，因为流量是 $O(n)$ 级别，所以一开始的最大流是 $O(n^3)$ 的。

同理，因为每个点的流量是 $O(1)$ 级别的，所以每次退流是 $O(n^2)$ 的，因此最后的时间复杂度是 $O(n^3)$。

CF1288F

Problem

有一张二分图，左边有 n_1 个点，右边有 n_2 个点，m 条边。

每个点可能有一种颜色 R 或者 B，也可能没有。

现在要给一些边染色，把边染成 R 要花费 r 的代价，把边染成 B 要花费 b 的代价，可以不染色。

要求对于每个颜色为 R 的点，与之相邻的边中 R 的边严格多于 B 的边。

对于每个颜色为 B 的点，与之相邻的边中 B 的边严格多于 R 的边。

求花费最小的方案，输出任意一种，无解输出 -1。

$1 \le n_1, n_2, m, r, b \le 200$。

CF1288F

Solution

考虑最小费用可行流。

对于原图中一条边 (u, v)，我们将它拆作两条边 $(u, v, 0, 1, R)$ 和 $(v, u, 0, 1, B)$。

如果它从左往右流，则这条边是红边；如果它从右往左流，则这条边是蓝边。

否则，如果它左右都不流，则这条边不染色。

CF1288F

Solution

考虑最小费用可行流。

对于原图中一条边 (u, v)，我们将它拆作两条边 $(u, v, 0, 1, R)$ 和 $(v, u, 0, 1, B)$。

如果它从左往右流，则这条边是红边；如果它从右往左流，则这条边是蓝边。

否则，如果它左右都不流，则这条边不染色。

对于一条红边，左部点付出流量，右部点收到流量；

对于一条蓝边，左部点受到流量，右部点付出流量。

CF1288F

Solution

对于左部的红点以及右部的蓝点 x，付出的流量应该大于收到的流量，因此连边 $(s, x, 1, \infty, 0)$；

对于左部的蓝点以及右部的红点 x，付出的流量应该大于收到的流量，因此连边 $(x, t, 1, \infty, 0)$；

对于一个无色点 x，因为它既可以付出流量，又可以受到流量，因此连边 $(s, x, 0, INF, 0)$ 和 $(x, t, 0, INF, 0)$。

然后跑最小费用可行流即可。

CF1615H

Problem

CF1615H

Solution

CF1781G

Problem

CF1781G

Solution