图论杂题选讲

CF1368G (*)

https://codeforces.com/problemset/problem/1368/G

显然对于一个格子，它如果想变空，那么要么一开始把它删掉，要么把覆盖它的骨牌移开。

那么从这个格子向骨牌将要覆盖的另一个格子连边，每个格子的出度为 0 或 1 。

看起来似乎变成了一棵基环树?

画一个 3*3 的有环的图，发现中间那个点非常不合法，于是大胆猜想没有环。

考虑一个环，可以证明它中间那片区域的面积一定是奇数，所以不合法。证明可以用神奇题解做法或画图+打表+猜结论。

所以图变成了一棵树。

黑白染色，那么一个骨牌的两个格子必然异色，所以从这两个格子出发到达的所有点也一定是异色的，所以交集为空。

所以大胆猜想能到达的所有点对都合法。

证明可以考虑：如果路径上会要求某一个骨牌被移动两次，那么直接改成删掉这个骨牌即可。

然后转 dfs 序，求矩阵并面积应该就可以了。

CF1458D

https://codeforces.com/problemset/problem/1458/D

考虑建图，在数轴上从 0 开始走路，见到 0 就往右走一步，否则往左走一步。走路的时候连有向边。最终走出一条欧拉路径。

考虑一次操作在干什么。区间 01 数量相同，即走出了一个环。一次操作就是把环上的边全部反向，走的顺序也反向。

然后大胆猜想，如果把图变成无向图，那么任意一个欧拉路径都是合法的。

证明可以考虑两条欧拉路径的第一个分叉点 x ，然后搞一下，就可以把分叉点的时间往后推。

字典序最小的欧拉路径可以贪心。

CF1466H (*)

https://codeforces.com/problemset/problem/1466/H

https://www.luogu.com.cn/paste/jfcee5mn

CF1477D

https://codeforces.com/problemset/problem/1477/D

转化一下就是要给一个无向图定向成 DAG，然后选取两个拓扑序使得对应位置相等的个数最少。

对于一个 DAG，显然答案的下界是这样的点 x 的个数：对于任意一个其他点 y，要么存在路径 $x \to y$，要么存在 $y \to x$。

那么对于一个无向图，下界就是度数为 $n-1$ 的点的个数。然后用构造的方法证明这个下界可以取到。

取补图，然后对每棵树分别构造即可。一种简单的构造方法是划分成若干个菊花。

CF986F

https://codeforces.com/problemset/problem/986/F

显然，满足条件当且仅当每个环的大小都是 k 的约数。

显然，我们可以经过调整使得每个环的大小都是 k 的质因子。

相当于是判断这些质因子做背包能不能凑出 n 。

我们还可以接着调整，使得较大的质数的系数小于最小的质数。设最小的质数是 p 。

所以有这么一个策略：用大的质数调整 n 模 p 的余数，当余数为 0 时就做完了。

调整有什么限制呢？不能在调整的时候把 n 弄到小于0。

所以可以建图：建 p 个点，表示模 p 的余数。每个大质数连 p 条边，表示调整，边权为大质数。

然后跑最短路即可。

特判 k 质因子少于 3 个。大于等于 3 个时 $p \leq 10^5$ ，可以通过。

CF1142E

https://codeforces.com/problemset/problem/1142/E

考虑没有红边的时候怎么做。

维护一个 root，和一个暂时不能从 root 走到的点集 S 。

每次从 S 里面拿出一个点 x，询问 (root,x) 。如果 $root \to x$ 就把 x 删掉，否则令 root=x，然后把 x 删掉。

有红边的时候，两点之间不再一定有绿边，所以需要想办法利用红边。

考虑把红边缩点，每个强联通分量只用一个代表点表示。维护所有入度为 0 的强联通分量的代表点组成的集合 S 。

任意选一个代表点作为 root，并从 S 里面删掉。如果此时 S 为空那么 root 就是答案。

每次仍然询问 (root,x)，但是此时不能简单地删掉 x，因为 root 经过绿边走到 x 之后就不能再往 x 所在的强联通分量走了。

理论上应该把 x 从图中删去，然后重新跑一遍 tarjan，加入新的代表点，但是这样很麻烦，而且复杂度也爆炸了。

更好的解决办法是保留这个强联通分量，仅仅删除 x，然后从里面随便再找一个点作为代表点。

如果这个强联通分量被删完了那么就把新的入度为 0 的强联通分量的代表点加入 S 中。

由于每次询问一定会删掉一个点，所以总询问次数最多是 n-1 。

CF1148G (*)

https://codeforces.com/problemset/problem/1148/G

大小**恰好**为 k 的限制比较恶心。

至少为 k：把每一个度数没有满的点拿出来即可，如果不够就取补集。

以下把图变成补图，即两点互质时有边。一个点的度数可以莫比乌斯反演求出来，复杂度 $O(2^c)$，其中 c 是质因子个数。

一种构造方法：每次拿出来两个互质的点，直到拿不出或是点数够。k 是奇数的时候稍微麻烦一点，可以先把一个度数大于 1 的点拿出来，并拿出两个和它相邻的点，凑成三个，然后再跑上面的构造，即可解决 k 为奇数的情况。

如果度数都小于等于 1，那么每一对点拿一个，再把拿出来一些孤点即可。

现在的问题就是怎样快速拿出两个互质的点，然后发现这东西不怎么能做……

正解：

还是先拿掉一个三元组，然后把此时度数为 0 的点删掉（如果删掉的点够多就直接输出），剩下来的点必定组成一个 unfair 集合，且这个集合加上三元组的大小 $\geq k$。

设 $f(r)$ 表示只考虑这个集合中的前 r 个点，那么度数不为 0 的点有几个。

我们可以二分出一个 c，满足 $f(c-1)+3<k\leq f(c)+3$。

容易发现前 $c-1$ 个点中恰好有 $f(c)-f(c-1)-1$ 个点只与 c 相连。我们只需要把前 c 个点中度数不为 0 的点拿出来，然后再把这些和 c 相连的点删掉一些（注意不能删完），就可以得到大小为 k 或 $k+1$ 的集合。

如果是 $k+1$ 怎么办呢？在三元组中删掉一个点即可。

CF1240F (*)

https://codeforces.com/problemset/problem/1240/F

这种题看着就觉得应该是构造一个方案使得所有边都在里面。

考虑拆点拆成二分图，对于原图的边 $(x,y),x<y$ ，变成新图的 $(x,y+n)$ 。

尝试给新图染色，使得每个点的极差不超过1。这样显然合并之后也还是合法的。

对于一个新图的点，若 $deg=xK+y\ (y<K)$ ，则再把它拆成 $x+1$ 个点。此时我们尝试让每个点连出去的边的颜色互不相同。

每次新加一条边 (x,y) 。如果有一种颜色可以同时满足x和y就直接加入，否则设x的一种合法颜色是1，y的一种合法颜色是2。

先让这条边的颜色为2，然后对y说，我要把这条边变成1。

那么y就会对原先它的1边的终点说，我要把这条边变成2。

以此类推，发现这样不会走出环。如果出环那么可以发现原来的方案不合法，这显然是矛盾的。

CF1268D

https://codeforces.com/problemset/problem/1268/D

这个题看起来没有什么思路，所以我们不妨想一想只操作一两次的方法。

关于竞赛图有三个性质：

1. 竞赛图缩点之后是一条"链"，满足前面的点都向后面的点有连边（这很显然）。
2. $n > 3$ 个点的强联通竞赛图存在大小为 $n-1$ 的强联通子图。
3. 一个竞赛图强联通当且仅当把点按出度排序后不存在 $k < n$ 使得前k个点的出度之和是 $\binom{k}{2}$ 。

性质二的证明：归纳证明n个点的强联通竞赛图存在 $[3, n]$ 的哈密顿回路。

性质三的证明：考虑缩点之后的最后一个块，里面的点的出度比其他点的出度都更小。

获得这么些性质之后可以发现：

1. 缩点之后有超过3个块，那么设开头为H，结尾为T，反转中间任意一个点x，对于其他任意一个点y，都有路径 $H \to y \to T \to x \to H$ ，所以最小操作1次。
2. 如果一个块X至少4个点，那么可以选一个点x出来使得反转之后它还是个块，然后有路径 $x \to (X-x) \to y \to x$ 或 $x \to y \to (X-x) \to x$ 。

只要满足这两条中的任意一条，就都可以枚举反转哪个点，然后用性质三判断。

否则，一定有 $n < 7$ ，直接暴力即可。

CF487E (*)

https://codeforces.com/problemset/problem/487/E

广义圆方树模板题。

建起圆方树后，令方点的权值是双联通分量中的最小值，那么(u,v)的答案就是路径(u,v)上的最小值。

然而这题还有修改，可以在每个方点维护一个$multiset$以支持。

但如果每次修改都暴力修改相邻的方点权值显然要挂，如何优化?

可以令方点的权值不包括自己的父亲，那么修改圆点时只需要修改自己的父亲即可。

查询时如果lca是方点那么还要与方点的父亲取$\min$。

CF718E

https://codeforces.com/problemset/problem/718/E

考虑两个点的最短距离，显然不会超过 16 。

设 $dis_{i,c}$ 表示从 i 走到 $s_x = c$ 中的任意一个点所用的最短时间；设 $dis_{p,q}$ 表示 $s_x = p$ 走到 $s_y = q$ 的最短时间。

那么容易发现 i, j 的最短距离即为 $min(|i-j|, min_c dis_{i,c} + dis_{j,c} + 1)$ 。

然而这东西也不是很好数。

$dis_{i,c}$ 有一个性质： $dis_{s_i,c} \leq dis_{i,c} \leq dis_{s_i,c} + 1$ 。

那么记 msk_i 为一个 8 维向量，每一维的值为 $dis_{i,c} - dis_{s_i,c}$ 。

于是 $dis_{i,c} + dis_{j,c} + 1 = dis_{i,c} + dis_{s_j,c} + msk_{j,c} + 1$ 。

对每一种字符的每一种 msk_i 开一个桶，那么就可以枚举 i, s_j, msk_j ，然后统计答案了。

$|i-j|$ 较小的直接暴力。

CF750H (*)

https://codeforces.com/problemset/problem/750/H

问题等价于两点之间最小割是否大于等于 2 。

设有障碍的格子为黑色。

转对偶图，然后人脑优化一下，变成这个样子：把每个格子看做点，相邻的黑点连边，问能否再染黑一个格子，使得左下和右上连通。

（所有靠着左边界和下边界的点都和左下的虚点连边，右边界和上边界的和右上的虚点连边）

我们先预处理出初始的所有连通块，并处理出两个连通块之间能否再染一个点使得连通。

那么可以发现此时再染黑的格子要么与某个 temporary 点相邻，要么使得某两个左下和右上的连通块连通。

此时左下的连通块最多包含 $O(k)$ 个原先的连通块，右上也一样，所以可以暴力枚举。

CF798E (*)

https://codeforces.com/problemset/problem/798/E

容易想到建图，然后跑拓扑排序。

显然需要把建图优化一下。

记 b_i 表示 i 被谁 mark 了。那么除了一条很 trivial 的边以外，其他边都是 $(i,j), 0<j<a_i, b_j>i, j\neq i$ 。可以用线段树找到这样的一个 j 。

我们把边反向，然后采用 dfs 的拓扑排序方法：从一个点开始，把它的所有出边都删掉，然后 dfs 它的前驱。

删边相当于在线段树中把自己删掉，找前驱也可以在线段树中找，于是就 $O(n\log n)$ 了。

CF843D (*)

https://codeforces.com/problemset/problem/843/D

数据范围中 q 这么小，猜想每次可以把每个点的最短路全都更新出来。

边权加的总和不大，所以也许我们可以换一种求最短路的方式。

考虑这么一种最短路算法： 与 dikstra 类似，但不用堆维护所有未进行松弛操作的点，而是加入一个桶里。即把 (x,d) 的二元组变成在 d 的桶里加入 x 。

这种方式的不足在于最短路很长的时候复杂度很爆炸。

然而我们每次修改的时候可以对图进行一些操作：

在知道原先的最短路的情况下，设 $w'(u,v)=dis_u+w(u,v)-dis_v+delta(u,v)$ ，然后，对 w' 求最短路，即可求出此时每个点的最短路与最初的最短路的差。

在不加 $delta(u,v)$ 时最短路都是 0 ，所以加完后不会超过 m ，于是更新的复杂度就可以做到 $O(n+m)$ 。

所以总复杂度 $O(q(n+m))$ 。

LOJ3057 (*)

https://loj.ac/p/3057

暴力很好想：把所有初始合法的点对丢进队列，然后暴力枚举两边往同色点走，$O(m^2)$ 。

注意到点数不多，所以这个思路可能是对的。所以考虑把边数砍下去，删掉一些无用的边。

把边分成连接同色点或异色点两种。

考虑同色连通块的作用，可以两边在里面乱跳，那么能否跳到同一个点呢？容易发现这和距离奇偶性和是否有奇环有关。对于二分图，只需要留一棵生成树，否则再加一个自环。

异色连通块也差不多，但它肯定是二分图，所以留一棵生成树。

然后边数就变成 $2n$ 就可以暴力了。

LOJ2493 (*)

https://loj.ac/p/2493

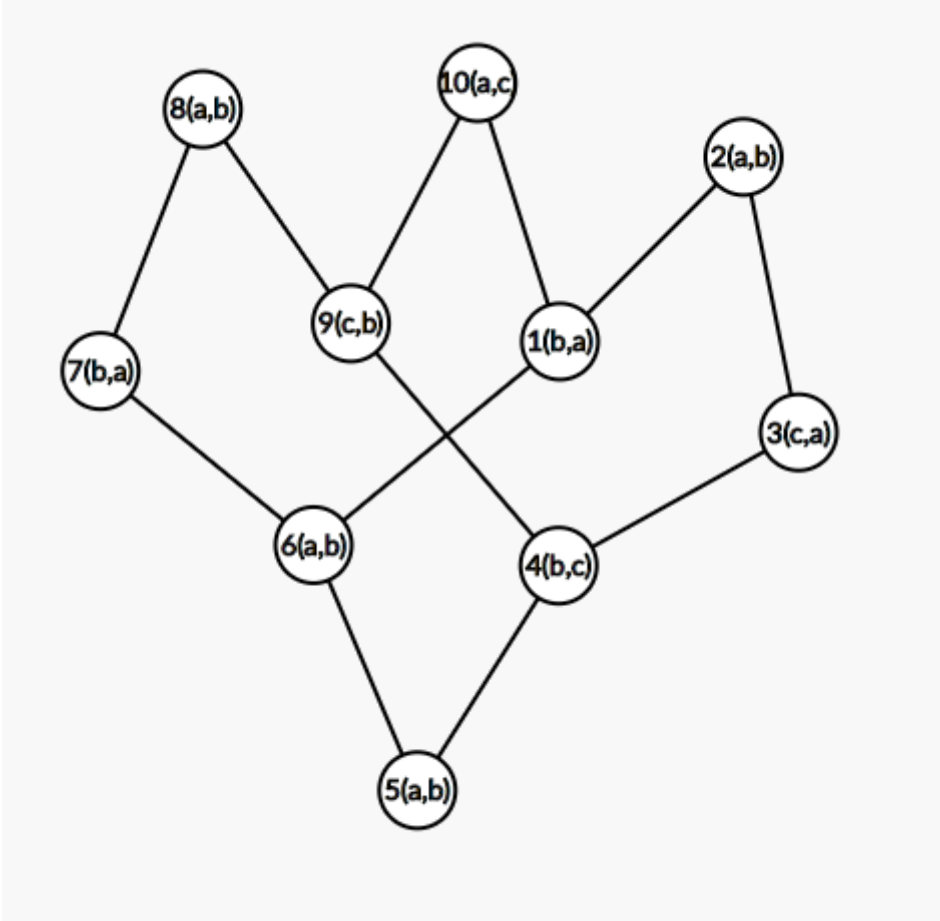
首先不是二分图的东西肯定可以被搞掉。

然后样例已经给了我们一种叉二分图的方法。总结一下就是：用一个环可以强制某个节点选某种颜色。

于是我们可以发现如果一个连通块内有两个不相交的环那么一定会被叉掉。

然后如果连通块内$m \leq n$，那么显然不能被叉掉。

然后如果一个连通块内$m \geq n+2$，那么有一种把它叉掉的方法：



于是就只剩$m=n+1$的情况了。此时可以抽象成两个度数为3的点，它们之间有三条路径。

手画一下可以发现长度$(1,3,3),(2,4,4)$的时候是可以被叉掉的，而长度更长的时候多半也是可以被叉的。

你发现别的都不怎么能被叉掉了，而“别的”就只剩$(2,2,\text{偶数})$了，就做完了。

CF1338E

https://codeforces.com/problemset/problem/1338/E

太神了，已经忘掉了，所以留作作业 /cy

AGC016D

容易发现异或是假的。设 $a_0 = \oplus_{i=1}^{n} a_i$ ，那么一次操作就是交换 a_0, a_i 。现在要用最少的交换次数把 a 变成 b 。

如果 a 里面的元素互不相同，那么容易想到在 $a_i = b_j$ 时连边 $i \to j$ ，会连出若干个环。最后答案就和点数、自环个数、环个数有关。

换一种建图，连边 $a_i \to b_i$ ，那么新图中的一个环可以对应回原图的一个环。新图的每个弱连通块显然都有欧拉回路，所以一个弱连通块就是一个环，很容易计算答案。

AGC051D (*)

我们会对有向欧拉图的欧拉回路计数，但是这里有两个问题，图是无向的，且起点终点相同的边是不可区分的。第一个问题很好解决，因为可以发现只有 $O(a)$ 种定向的方法。

对于第二个问题，可以考虑魔改欧拉回路计数问题的解法。原解法是任取一棵内向树，然后任意排列其他的边。这里还是任取一棵生成树，但把排列变成组合即可。

AGC031F

先把过程反过来，变成从 T 到 S，维护一个 sum，每走一步就 $sum:=2sum+w$，需要走到 S 时 $sum=R$。所以状态数变成 $O(nM)$ 了。

考虑最小的 $t\,(t>0)$ 使得 $2^t=1\pmod M$。如果 t 是奇数，那么在一条边上来回走 t 次就可以在不改变 sum 的情况下移动，所以本质不同的状态数只有 M 种，可以暴力。

否则，走 t 步会走回自己。考虑一个点旁边的两条边 u,v，先在 u 上走 2 步，再在 v 上走 $t-2$ 步，就可以收获 $sum:=sum+3(u-v)$。

考虑从 (x,sum) 走到 $(y,2sum+w)$，然后走 t 的倍数步回到 $(y,2sum+w+?)$，然后再反复横跳，就可以走到 $(x,sum+\frac{1}{2}?)$。所以 y 能得到的额外贡献 x 也都可以得到。

综合上面两行，可以发现，可以把所有共顶点的 $3(u-v)$ 取 $\gcd$，并与 M 取 $\gcd$ 得到 $\gcd(3(u-v),M)$。

AGC008E

建图，对于某个合法方案p，连$i \to p_i$，那么一定可以组成若干个环。

再连$i \to a_i$，那么这条边在环上一定只有两种情况：和原来的边一样，或是跳过了一个点，到了下一个点。

如果所有边都一样，那么显然还是原来那个环。

如果所有边都跳了一下，那么如果环是奇环就还是一个环，否则就会分裂成两个环。

如果有些跳有些不跳，那么就变成了一棵特殊的基环树，这里的“树”其实是一条链。

现在我们手上有$i \to a_i$的图，考虑如何复原出$i \to p_i$的图。

对于长度相等的简单环可以一起考虑，分自己成一个环和与前面环合并来讨论。

对于基环树，要把那些多出来的链塞回去。可以去看这里。（litble Orz）

AGC010E

可以发现，定下初始序列之后，不互质的数的相对顺序不会改变。

我们把不互质的数连起边来，成为若干个连通块，现在单独考虑每个连通块。

如果要最小化每个连通块的顺序的字典序怎么做呢？显然第一个可以放最小的，但这也就限制了第二个必须与第一个有边相连，同时也限制了后面的……但是没有关系，只要每次选与自己有边的、编号最小的往后dfs即可。

假设我们固定了每个连通块的顺序呢？可以发现后手一定会贪心地每次取最大的往前填，所以只要最小化每个连通块字典序即可。

AGC045E

没啥思路，于是建图，连边 (A_i, B_i, C_i) ，其中 C_i 为边权。

缩点，那么会得到一个 DAG 。先不考虑最小化步数。

考虑对于一个强连通块（如果非简单环），一定可以做到每一个球都一步走完：无脑走，最后如果出现了简单环，那么可以在这个环上的另一条边走掉之前把这一整个环给走掉。

所以对于一个弱连通块，只要不是简单环，都可以使得所有不在目的地的球一步走到。（注意由于保证了每个点都有入度，所以非简单环时球数一定大于点数）

对于一个简单环，需要一个球冲进来，把一整个环走掉，然后再把这个球送走。

然后就是大力分类讨论了……可以把弱连通块分为三种。

- 仅自环。（只有一条边）
- 仅简单环。（同样不能有重边）
- 其他所有弱连通块。

设 “费用” 表示除了 $\sum[a_i \neq b_i]$ 之外的步数。为了计算费用可能会把多余步数贡献到跳的球或被跳到的点上。

以下基本不再区分球和点。

对于一类块，如果有点进来，那么它可以跳出去进行操作再跳回来，可以贡献 $c-1$ 次移动，额外费用是 2 （自己跳回来一步，别人跳过来一步）。

对于二类块，显然必须有点进来，那么一个点可以在走到 B_i 之前先跳出去进行操作，然后直接跳到 B_i ，所以一共贡献 $\sum(c_i-1)$ 次操作，额外费用是 1 （别人跳过来一步）。

对于三类块中的非自环，可以贡献 c_i-1 次操作，额外费用是 0 。

对于三类块中的自环，可以贡献 c_i-1 次操作，额外费用是 1 （自己跳回来一步）。

显然应该优先用三类块中的非自环球，然后用二类的球，然后用三类的自环球，最后用一类球。

设一类用了 X 个，二类有 Y 个，那么需要 总贡献 $\geq X+Y$ ，额外费用为 $2X+Y+\text{三类块中用的自环个数}$ 。并且如果 $X+Y\neq 0$ 那么三类贡献必须非 0 。

瞎搞搞即可。

AGC029F

大胆猜想某一些条件合在一起就是充要条件。经过颅内随机游走，可以想到：任取一个点集 $S\,(|S|\neq 0)$ ，完全包含在 S 内的集合个数不能超过 $|S|-1$ ，且这是充分条件。

> 证明：
>
> （其实直接看后面的转化和构造就可以了）
>
> 设 $f(S)$ 表示包含在 S 内的集合个数。
>
> 如果这个条件满足，那么任取 $f(S)=|S|-1, f(T)=|T|-1, S\cap T\neq\varnothing$ ，会有 $f(S\cap T)=|S\cap T|-1$ 。
>
> 现在找到最小的 $f(S)=|S|-1, |S|\neq 1$ ，在里面任意用一个集合 E 连接一条边 (u,v) 。如果用掉之后不合法，那原来一定存在一个 $u,v\in T, f(T)=|T|-1, E\nsubseteq T$ ，所以 $f(S\cap T)=|S\cap T|-1$ ，与 S 最小矛盾。
>
> 重复这个过程直到只剩一个点即可。

这个条件等价于任取 $k\,(k\neq 0)$ 个 E ，它们的并集的大小要超过 k 。也就等价于任意删掉一个点后 V 和 E 存在完美匹配。

从 E 往 V 跑一个匹配，会剩下一个孤立点。从这个孤立点开始跑一个增广路树，如果能遍历到整个 V 那么就容易构造出解，否则任意删掉一个不能被遍历到的点之后就没有完美匹配了。

CTS2020 D2A (*)

先把完美消除序列求出来。两个子集在完美消除序列中的子序列显然就是对应的导出子图的完美消除序列。

因为最大团的大小是 $2k$，容易想到尝试构造使得两边的大小都是 k。

事实上这很容易：在序列上贪心染色，把这个点的颜色染成前面和它有边的数量较少的颜色即可。

不难证明这个构造的正确性。