AGC036D

给定 n 个点的有向图，其中：

- $\forall 1 \le i < n$，有边 $i \to i+1$，边权为 0，这些边不可删除。
- $\forall 1 \le i, j \le n, i \ne j$，有边 $i \to j$，当 $i < j$，边权为 -1，否则边权为 1，删掉这条边需要支付 $A_{i,j}$。

求最小的花费，使得图中没有负环。

$3 \le n \le 500, 1 \le A_{i,j} \le 10^9$。

solution

考虑 `Johnson` 算法。发现没有负环等价于可以给每个点赋一个势 h_i，使得对于所有边 (u,v,w)，$h_v-h_u+w\geq 0$。

那么就是找到一组势 h_i，如果边 (u,v,w) 使得 $h_v-h_u+w<0$ 则需要删掉这条边，求最小代价。

由于存在 $(i,i+1,0)$ 的边，所以 $h_{i+1}\geq h_i$。

考虑 $i<j$，如果有 $h_j-h_i-1\geq 0$，则不需要支付贡献。

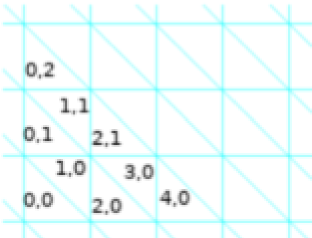
考虑 $i>j$，如果 $h_j-h_i+1\geq 0$，则不需要支付贡献。

那么考虑相当于把点划分成若干段，第 i 段势为 i，那么我们需要支付同一段里 -1 的边的代价，以及势相差 2 以上的段之间 1 的边的代价。

朴素的 $O(n^3)$ `dp` 即可。

CF1446E

考虑下面这种平面：



即 (i,j) 的邻居为：

- $(i+1,j)$
- $(i-1,j)$
- 当 i 为偶数时，$(i+1,j-1)$；否则，$(i-1,j+1)$。

现在有若干个点被染黑了，每个时刻，你可以进行下面恰好一种操作执行：

- 当存在一个白点拥有至少两个黑邻居时，你可以把它染黑。
- 当存在一个黑点拥有至少两个白邻居时，你可以把它染黑。
- 如果以上两种点都不存在，结束。（**注意**，必须上面两种操作都执行不了才能结束）

如果可以使得结束时平面上至少有一个黑点，则输出 `SICK`。

否则，输出结束的最晚时刻，由于答案可能很大。

$1\leq n\leq 250000$，其中 n 是一开始染黑的点的数量，$0\leq x_i,y_i\leq 500$，其中 (x_i,y_i) 为初始染黑的点的坐标。

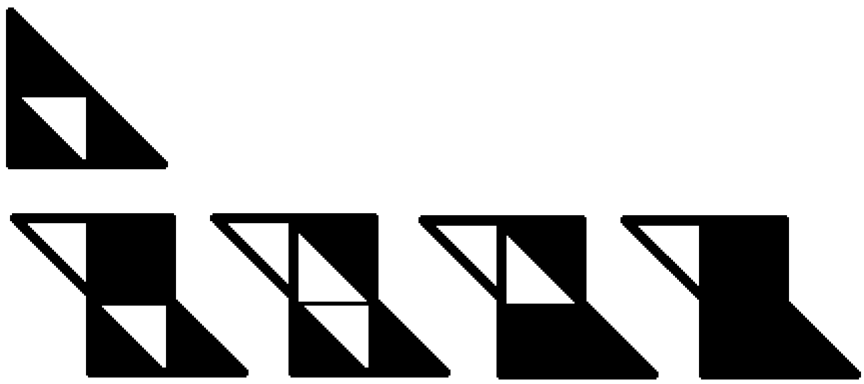
solution

考虑我们可以尽量把所有能变黑的格子变黑，然后把相邻的黑格子连边，如果最后有环，那么就永远不可能治愈，否则一定会在有限步内治愈。

设 P 为相邻的不同色格子对数，那么每次操作一个格子，如果是恰好两个邻居和它异色，那么 P 会减一，否则会减三，我们尝试从这方面来考虑最长治愈时间。

对于每个连通块，最后一步操作一定是让 P 减三的，所以先把 P 减去连通块数乘二。

我们把一开始就是黑的格子称为 1 类，后来被染黑的格子称为 2 类，考虑我们把 2 类全部染黑后，接下来除了最后一步，通过每次染白叶子，一定可以恰好把 P 减一。考虑哪些树可以每次把 P 减一来染黑，那么发现除了下面这种情况都可以：



证明：对于度数为 2 的 2 类，显然染它只需要一步，对于度数为 3 的 2 类就考虑下面四幅图的变化过程。

那么只需要特判这种连通块，每有一个这种连通块都要把 P 减二。

复杂度 $O(W^2)$。

TopCoder-14750 SRM726 Div1C

给定一个二分图，左部有 n 个点，右部有 m 个点，左部第 i 个点向右部 $[l_i, r_i]$ 的点连边，点权为 a_i，右部点点权为 0，求最大权匹配。

$1 \le n \le 10^7, 1 \le m \le 2000$。

solution

考虑按点权从大到小考虑左部点尝试加入匹配，如果加入失败可以知道它接下来就没有用了，暴力实现匈牙利增广路是 $O(nm^2)$ 的。

考虑匹配只会变化 m 次，每次都跑一次增广是非常差的，我们可以令每个右部点向所有能走到它的左部点连一条有向边，每个左部点向它的匹配点连一条有向边，那么我们从所有不在匹配的右部点开始 `bfs`，把到达的点全部标记，那么加入一个左部点 i 时，只需要检验右部 $[l_i, r_i]$ 中有没有被标记的点，如果有就跑增广路并重构，复杂度 $O(n+m^3)$。

考虑线段树优化建图，复杂度 $O(n+m^2\log m)$。

[THUSCH2017] 巧克力

给一个 $n\times m$ 的矩阵以及常数 k，矩阵上每个点有价值 $a_{i,j}$ 和颜色 $c_{i,j}$。你要选出一个四连通块，使得不存在一个点的颜色是 -1，至少有 k 种不同的颜色，且连通块包含点数最少。在此基础上，你还需要令这个四连通块里所有点的价值的中位数最小。

$1\le nm\le 233, 1\le k\le 5, c_{i,j}=-1\vee 1\le c_{i,j}\le nm, 0\le a_{i,j}\le 10^6$。

solution

二分中位数 x，把价值比 x 小的赋值为 1000，比 x 大的赋值为 1001。那么我们只需要求出价值之和最小的至少包含 k 种颜色的连通块即可。

考虑给每种颜色都随机附一个 $[1,k]$ 里的权值，把问题变为求价值和最小的至少包含 k 种权值的连通块，用斯坦纳树解决复杂度为 $O(nm3^k)$。最优解中每个颜色赋到的权值互不相同的概率为 $\frac{k!}{k^k}$，进行 200 次左右正确概率就大于 $1-4\times10^{-4}$ 了。

总复杂度 $O(Tnm3^k\log W)$，其中 W 为价值值域，T 为随机次数。

PTZ winter 2020 Day2 K

给定一个 $n\times m$ 的矩阵和常数 a,b,c，初始时矩阵里每个格子颜色都为白色，你需要给这些格子染上黑白色，有若干个格子被钦定最终为黑色，其他所有格子被钦定最终为白色。你可以以任意顺序使用如下两种操作任意次来染色：

- 选择若干个排成一条平行于坐标轴的线段的连通的格子，然后选择把它们都染成黑色或都染成白色，假如选择了 l 个格子，花费为 $al+b$。
- 选择一个格子，把它染成黑色或白色，花费为 c。

你还需要满足：任何一个格子不能被染两次色，一个被染成白色的格子不能再被染色。

求满足颜色要求的最小花费。

$1\le n,m\le 40,0\le a,b,c\le 40,c\le a+b$。

solution

考虑到我们最优解中，每个格子至多被竖着的线段覆盖一次，横着的同理。

我们可以设 $Hb_{i,j},Vb_{i,j},Hw_{i,j},Vw_{i,j}$，分别为是否被黑/白的竖着/横着的线段覆盖了。

以黑色竖线为例，花费是 $a\sum Hb_{i,j}+b\sum Hb_{i,j}\overline{Hb_{i+1,j}}$。

然后考虑图形的限制，对于一个要是黑色的点，花费就是 $c\overline{Hb_{i,j}}\,\overline{Vb_{i,j}}+\infty(Hw_{i,j}+Vw_{i,j})$，后面是因为如果染了白色就染不上黑色了。

对于白色的点，花费为 $c(Hb_{i,j}\overline{Vw_{i,j}}+Vb_{i,j}\overline{Hw_{i,j}})+\infty Hb_{i,j}Vb_{i,j}$，后面是因为一个格子只能染两次。

这些可以建成最小割模型求解，具体的，假设 S 是源点，T 是汇点，对每个变量建一个点。

把 $Hw_{i,j}$ 和 $Vb_{i,j}$ 的意义取反，所有式子都变成 $cX\overline{Y}$ 的形式。

$\sum u_i\overline{v}_i c_i$ 权值的处理

假如有若干个 `01` 变量 x_i，然后我们要最小化一个和式，和式的每一项都形如 $x_{a_i}\overline{x_{b_i}}c_i$ 或 $x_{a_i}c_i$ 或 $\overline{x_{a_i}}c_i$。

我们把每个变量看作点，当点与 S 不连通，我们认为它为 1，否则为 0。那么 $x_{a_i}c_i$ 我们看作 (S,a_i,c_i)，同理 $\overline{x_{a_i}}c_i$ 看作 (a_i,T,c_i)。然后 $x_{a_i}\overline{x_{b_i}}c_i$ 就是 (b_i,a_i,c_i)。那么和式的最小值变成了 S 和 T 的最小割。

复杂度 $O((nm)^3)$，网络流玄学。

gym103860H

给定一个 n 个点 m 条边的二分图，求有多少个点对，满足删去这两个点最大匹配不变。

$2\le n\le 2\times 10^5,0\le m\le 2\times 10^5$。

solution

先跑一个最大流，考虑删去一个点后最大流不增，所以点对里两个点都满足删去后最大流不变。

先考虑左部点，如果边 (S,i) 没流，则删去点 i 不影响最大流。否则，假如你能在残余网络中找到一个包含 (i,S) 这条边的增广环，则你可以在不影响最大流的情况下把 (S,i) 这条边的流量退掉，然后就可以删去 i。

新建虚点 i'，初始建边时把边 (S,i) 拆成 $(S,i'),(i',i)$，那么上面的条件相当于 S 可以在残余网络上走到 i'。右部点也类似。

如果你选择的点对不在同一侧，那么它们的选取是独立的，因为 S 和 T 已经割开了，所以直接对每一侧统计满足上述条件的点数再相乘即可。

假如选择的点对在同一侧，以左侧为例，假如点对为 x,y，则这个点对合法当且仅当能选择两条边不交的 S 分别到 x',y' 的路径。那么把残余网络剩下的边也像上面一样拆开，则只需要 x',y' 在支配树上的公共祖先都不是虚点即可，（如果虚点支配了一个点 u，说明 S 到 u 的路径必须经过这个虚点对应的边）。建完支配树在支配树上统计答案即可。

复杂度 $O(m\sqrt{n}+m\log m)$。

PTZ summer 2021 Day7 E

给定点数为 n，边数为 m 的无向图，每个点有危险值 c_i，阈值 t_i，价值 s_i。

记 f_u 为，你从 u 出发，所能到达的价值最高的点的价值。假如你现在所在的点为 x，则你能去点 y 当且仅当有边 (x,y) 且你从出发到现在经过的所有点的危险值的最大值小于等于 t_y。

你需要对每个 $1\le i\le n$ 求出 f_i。

$1\le n,m\le 2\times 10^5, 1\le c_i,s_i,t_i\le 10^9, c_i\le t_i$。

solution

我们考虑 $f(x,u)$ 表示从 u 出发，带有带有风险等级 x，最大可以到达的 s，考虑如何对于所有 $c_u \le x$ 的 u，求出 $f(x,u)$。

考虑那些 $t_i < x$ 的城市你永远都到不了，删去。那么剩下的城市里，如果有边 (u,v)，且 $c_u \le x \wedge c_v \le x$，那么它们从 x 开始的话可以互相到达而不增加，我们把它们合并，设 S_u 为 u 所在可以互相到达的集合。假如 $c_u \le x \wedge c_v > x$，那么我们认为 u 向 v 连了一条单向边，记为 $u \to v$。那么我们知道 $f(x,u) = \max(\max\{s_v \mid v \in S_u\}, \max\{f(c_w, w) \mid v \in S_u, v \to w\})$。

考虑一条边的两个端点可以互相到达的时间是一个段 $\max(c_u, c_v) \le x \le \min(t_u, t_v)$，成为单向边的时间为 $c_u \le x \le \min(t_u, c_v - 1)$（假设 $c_u < c_v$）。

那么考虑分治解决问题。

我们在分治的过程中，记录下在这个时间段可能成为双向边的边和可能成为单向边的边，当一条边在这这个时间段一直会成为单向边或一直会成为双向边的时候就用并查集维护一下。记得先递归右边的先得到那些 $f(c_w, w)$，才能更好地维护单向边的转移。退出分治区间地时候要撤销连边所以需要可撤销并查集。

复杂度 $O(m \log^2 n)$。

hdu 6691

你有 n 个点和一个正整数 k，$\forall i \neq j$，有 p_0 的概率 i 和 j 之间没有连边，对于 $1 \le l \le k$，有 p_l 的概率 i 和 j 之间有一条边权为 l 的边，保证 $\sum\limits_{i=0}^{k} p_i = 1$。

对于 $t = n-1, n, \ldots, (n-1)k$ 分别求出，整张图连通且最小生成树为 t 的概率，模质数。

$1 \le n \le 40, 1 \le k \le 4$。

solution

考虑 `kurskal` 求生成树的过程，发现加入了 $\le i$ 的边后，点之间的连通性是固定的。

因为所有点都是等价的，我们假设 $f_{i,j}$ 为加入了 $\le i$ 的边后，某个特殊点所在的连通块大小为 j 的概率，此处我们假设这个特殊点是连通块里编号最小的点。

考虑转移，可以得到 $f_{i,j}=\sum\limits_l f_{i-1,l}g_{i,l,j-l}\binom{j-1}{l-1}$。其中 $g_{i,j,k}$ 表示 $\le i-1$ 的边加入后，有一个大小为 j 的连通块，加入 $\le i$ 的边后，这个连通块大小变成 $j+k$ 的概率。

考虑求 g，发现大小相同的连通块是等价的，所以考虑枚举加入的连通块的大小 l 和数量 m 一起转移：

$$\frac{g'_{i,j,k}(t_{1,i,j,l}f_{i,l})^m(\prod\limits_{p=0}^{m-1}t_{0,i,j+pl,l})(\prod\limits_{p=1}^{m}\binom{j+pl}{l})}{m!}\to g_{i,j,k+ml}$$

$t_{0,i,j,k}$ 表示对于两个大小分别为 j 和 k 的连通块，它们之间要么没有连边要么连边的边权 $>i$，即为 $(p_0+\sum\limits_{j>i}p_j)^{jk}$。

$t_{1,i,j,k}$ 表示对于两个大小分别为 j 和 k 的连通块，它们之间要么没有连边要么连边的边权 $\ge i$，且至少有一条边权 $=i$，即为 $t_{0,i-1,j,k}-t_{0,i,j,k}$。

那么我们通过 $O(kn^3\log n)$ 的递推得到了 g，再通过 $O(kn^2)$ 推出了 f。

但此时我们算出来的 $f_{k,n}$ 其实是最后整张图连通的概率，所以我们考虑加一维状态来记录当前连通块的最小生成树的权值。但这一维的范围是 $O(nk)$ 的，那我们递推 f 的复杂度变成了 $O(k^3n^5\log n)$，很爆炸。

但我们可以把 $f_{k,n}$ 看作多项式，多项式的第 i 项的系数即为此时最小生成树为 i 的概率。我们发现这是一个不超过 $(n-1)k$ 次的多项式，并且转移时永远是第 i 项和第 j 项贡献给 $i+j$ 项。那么我们不直接维护多项式，而是代入 $x=n-1,\dots,(n-1)k$，并计算此时多项式的点值，此时发现只需要枚举 x，其他与上面式子类似，复杂度缩回了 $O(k^2n^4\log n)$，似乎还有一个 $\frac{1}{6}$ 的常数，很稳。最后通过这些点值高斯消元或者拉格朗日插值 $O(n^3k^3)$ 或 $O(n^2k^2)$ 还原多项式系数即可。