图论选讲

张怀沙

Tsinghua University

2025 年 4 月 28 日

主要是从题目体现模型与技巧，所以题目并不会太难，且有部分经典的题目。
如果遇到做过的题，可以重新想一遍。

建图意识

图本质上是描述事物之间的二元关系。
当题目中出现与此相关的条件时，应该引起注意。

PA 2014 Kuglarz

先上点开胃小菜。

题目描述

魔术师的桌子上有 n 个杯子排成一行，编号为 $1, 2, \ldots, n$，其中某些杯子底下藏有一个小球，如果你准确地猜出是哪些杯子，你就可以获得奖品。
花费 c_{ij} 元，魔术师就会告诉你杯子 $i, i+1, \ldots, j$ 底下藏有球的总数的奇偶性。
采取最优的询问策略，你至少需要花费多少元，才能保证猜出哪些杯子底下藏着球？

Sol

设前缀和为 S_i，花费 c_{ij} 元相当于知道 S_{i-1} 与 S_j 关系，连一条边。
最后发现就是求最小生成树。

差分约束模型

解决一类特殊的线性规划问题。

利用最短路中三角形不等式 $d_j - d_i \le e_{ij}$ 来表现二元不等式关系。

即从 i 向 j 连权为 w 的边，则 $d_j \le d_i + w$。

设最后每个变量为 a_i，实际上由于 $a_1 + d_i \ge a_i$ ，所以其实最后求出的 d_i 相当于变量的上界。

平凡的技巧：利用取负数和取对数等操作，可以进一步扩展为两变量的和、积、商的限制。

THUSC 2018 回合战略

题目描述

圆上均匀分布 $2n$ 个点，顺时针从 0 开始编号。A 控制编号为奇数的点，并在这些点间连了 m 条带权边。B 控制编号为偶数的点，他需要在这些点上连若干条带权边，使得对 A 的所有边，其边权小于等于与其相交的 B 的边的边权之和。最小化 B 连边的边权之和。

$n \leq 2000, m \leq 4000$。

Sol

如果直接考虑所有可能的连边各自的边数，这是经典线性规划问题，但其变量个数为 $O(n^2)$ 的，无法接受。

转换思路，转而考虑端点。一个显然的必要条件是：取出 B 连的所有边的端点 (重复的计重数)，对 A 连的权值为 w_i 的边，其两侧的端点数之和应该分别 $\geq w_i$。

仔细思考可以发现这是充分条件，证明可以考虑构造后调整。于是问题就转化为了指定若干个区间 $\geq w_i$，转成前缀和后差分约束即可。

团与独立集

由于团、独立集相关的问题往往为 NP-Hard 问题，因此一般情况下给出的图有特殊性质。

同时，两者为在取补图意义下为对偶关系，这常常作为突破口。

Unknown Source

题目描述

给定 n 个点、m 条边的无向图，试将所有顶点分为两个集合，使得两个点集的生成子图均为团。求两个集合大小之差的最小值。

$n \le 1000, m \le \frac{n(n-1)}{2}$。

Sol

想到第一步就结束了。

取补图，发现相当于构造二分图。考虑每个联通块，不是二分图就无解，否则可以求出两侧点的数量，背包 DP 即可。

优化连边

有时候，直接根据题目建出的图规模往往过大，这时需要减小图的规模。

一般情况下，需要结合使用的图相关算法的性质来优化，这部分内容也将分散在后面提及。

基础

求最短路的基础算法应该不用多说。

把握三角不等式。

重点是建模，以及某些特殊图的最短路。

基础

思考题：

- 01 最短路推广：边权 $\le k$ 的最短路，有没有更优秀的算法？
- 如果还有 c 条边权任意的额外边呢？

答案：

- 开 $k+1$ 个队列，复杂度 $O(kn+m)$。
- 再开一个优先队列，复杂度 $O(kn+c\log n+m)$。

基础

思考题：

- 01 最短路推广：边权 $\leq k$ 的最短路，有没有更优秀的算法？
- 如果还有 c 条边权任意的额外边呢？

答案：

- 开 $k+1$ 个队列，复杂度 $O(kn+m)$。
- 再开一个优先队列，复杂度 $O(kn+c\log n+m)$。

CERC 2012 Farm and Factory

题目描述

给定 n 个点、m 条边的带权无向图，现要求增加一个点并向原来 n 个点各连一条无向边，使得任意点到点 $1, 2$ 的最短路可以不经过新点。最小化这 n 条边边权之和。

$n \le 10^5, m \le 3 \times 10^5$。

Sol

充分利用最短路性质。

设 $d_1(i), d_2(i)$ 代表 i 到 $1, 2$ 的最短路，A, B 为新点到 $1, 2$ 的最短路，w_i 为新点向 i 连的边，则有如下必要条件：

- $w_i + A \geq d_1(i)$
- $A \leq w_i + d_1(i)$

容易证明这也是充分条件。

故 $w_i \geq |A - d_1(i)|$。对 2 号点同理，有 $w_i \leq |B - d_2(i)|$。我们需要最小化 $\sum w_i$，相当于要确定 A, B，使得 $\sum \max(|A - d_1(i)|, |B - d_2(i)|)$。

这可以直接做，更好的方法是切比雪夫距离转曼哈顿距离后两维分开算。

Sol

充分利用最短路性质。

设 $d_1(i), d_2(i)$ 代表 i 到 $1, 2$ 的最短路，A, B 为新点到 $1, 2$ 的最短路，w_i 为新点向 i 连的边，则有如下必要条件：

- $w_i + A \geq d_1(i)$
- $A \leq w_i + d_1(i)$

容易证明这也是充分条件。

故 $w_i \geq |A - d_1(i)|$。对 2 号点同理，有 $w_i \leq |B - d_2(i)|$。我们需要最小化 $\sum w_i$，相当于要确定 A, B，使得 $\sum \max(|A - d_1(i)|, |B - d_2(i)|)$。

这可以直接做，更好的方法是切比雪夫距离转曼哈顿距离后两维分开算。

建图——观察性质

直接建出的图，可能需要优化连边以减小图的规模。
这时往往利用贪心等思想，分析不可能的路径，进而缩小图的规模。

Code+ #4 最短路

不算困难的题目。

题目描述

图中有 n 个点，从 i 走到 j 费用为 $C \times (i \text{ xor } j)$。此外还有 m 条额外单向边。求单源最短路。
$n \le 10^5, m \le 10^6$。

Sol

肯定需要优化连边。

看到位运算想到拆位，考虑从 i 走到 j，每次翻转 1 位、一步一步走的费用，不小于直接花费 $C \times (i \mathrm{xor} j)$ 到 j 的费用，且能取等。

然后就简单了，i 向 $i \text{ xor } 2^k$ 连边 $C \times 2^k$ 即可。

建图——数据结构优化

另一种常见的情况是一个点向某个范围连边。可以尝试数据结构处理。
典型的有线段树、K-D Tree、主席树、CDQ 分治等。

NOI 2019 弹跳

题目描述

$w \times h$ 的网格中有 n 个点，m 个装置，每个装置可以从某个点到一个矩形范围内的任意点，花费为 c_i。求单源最短路。
$n \le 70000, m \le 150000$。

Sol

虽然已经被讲烂了，但还是要拿出来讲。

首先点到矩形连边，想到的肯定是诸如树套树、K-D Tree 之类的高维数据结构。直接这么做，堆实现的好的话（如配对堆）或者像之前所说单独处理边权为 0 的边，复杂度为 $O(n\log n + m\sqrt{n})$ 或 $O(n\log^3 n + m\log^2 n)$。

有没有更优秀的做法?

考虑到每个点向外的边权种数总共只有 m 种。我们可以每次不是取出 dis_i 最小的，而是 $dis_i + c_j$ 最小的，这样我们的操作就从更新所有出边通向的点变成了找到矩形内的所有点并删除。**这也是一个常用的转化。**

这是个经典问题，直接以 x 这一维建线段树，每个结点存 set 即可做到 $O(n\log^2 n)$。一个更好的做法是线段树的结点上预先把点的坐标按 y 排好序，预处理询问矩形的坐标的 `lower_bound`，这可以利用扫描线实现 $O(n\log n)$。用并查集维护查找后面第一个没有被删除的点及删除点，总复杂度 $O(n\alpha(n)\log n)$。

Sol

虽然已经被讲烂了，但还是要拿出来讲。

首先点到矩形连边，想到的肯定是诸如树套树、K-D Tree 之类的高维数据结构。直接这么做，堆实现的好的话（如配对堆）或者像之前所说单独处理边权为 0 的边，复杂度为 $O(n\log n + m\sqrt{n})$ 或 $O(n\log^3 n + m\log^2 n)$。

有没有更优秀的做法？

考虑到每个点向外的边权种数总共只有 m 种。我们可以每次不是取出 dis_i 最小的，而是 $dis_i + c_j$ 最小的，这样我们的操作就从更新所有出边通向的点变成了找到矩形内的所有点并删除。**这也是一个常用的转化。**

这是个经典问题，直接以 x 这一维建线段树，每个结点存 set 即可做到 $O(n\log^2 n)$。一个更好的做法是线段树的结点上预先把点的坐标按 y 排好序，预处理询问矩形的坐标的 `lower_bound`，这可以利用扫描线实现 $O(n\log n)$。用并查集维护查找后面第一个没有被删除的点及删除点，总复杂度 $O(n\alpha(n)\log n)$。

ARC165F Make Adjacent

题目描述

给定长度为 $2n$ 的序列 a，满足 $[1,n]$ 每个数恰好出现两次。
每一次操作可以交换相邻的两个数，询问最少多少次操作可以使得序列 a 满足 $a_{2i-1}=a_{2i}\forall i=1\dots n$。
在保证为最小操作次数的前提下，输出所有可以得到序列中字典序最小的那一个。
$n\le 2\times 10^5$。

同样考验图论建模的能力。

Sol

类比逆序对，我们考虑数 i 和数 j 的相对顺序，设它们出现的位置为 i_1, i_2, j_1, j_2 $(i_1 < i_2, j_1 < j_2)$。

如果 $i_2 < j_1$，不需要交换，且最终序列一定是 i 在 j 前；若 $i_1 \le j_1 \le i_2 \le j_2$，需要 1 次交换，最终序列也是 i 在 j 前（要保证交换次数最小）；如果 $i_1 \le j_1 \le j_2 \le i_2$，需要 2 次交换，但这时最终序列 i, j 的顺序可以随意。对称的情况同理。

如果我们把所有诸如 $i < j$ 的限制在图上建出来，那么这就是个 DAG，需要求其字典序最小的拓扑序，这是经典问题，直接优先队列维护即可。

如何优化连边？对所有 i，我们需要找到所有 j 满足 $i_1 < j_1, i_2 < j_2$。

法 1：如果直接排序后扫描线，在树状数组上连后缀，这会导致后来加入树状数组的点会被前面的点连上。可持久化线段树即可。

法 2：使用 CDQ 分治。

Sol

类比逆序对，我们考虑数 i 和数 j 的相对顺序，设它们出现的位置为 i_1, i_2, j_1, j_2 $(i_1 < i_2, j_1 < j_2)$。

如果 $i_2 < j_1$，不需要交换，且最终序列一定是 i 在 j 前；若 $i_1 \le j_1 \le i_2 \le j_2$，需要 1 次交换，最终序列也是 i 在 j 前（要保证交换次数最小）；如果 $i_1 \le j_1 \le j_2 \le i_2$，需要 2 次交换，但这时最终序列 i, j 的顺序可以随意。对称的情况同理。

如果我们把所有诸如 $i < j$ 的限制在图上建出来，那么这就是个 DAG，需要求其字典序最小的拓扑序，这是经典问题，直接优先队列维护即可。

如何优化连边？对所有 i，我们需要找到所有 j 满足 $i_1 < j_1, i_2 < j_2$。

法 1：如果直接排序后扫描线，在树状数组上连后缀，这会导致后来加入树状数组的点会被前面的点连上。可持久化线段树即可。

法 2：使用 CDQ 分治。

同余最短路

核心在于划分为同余等价类。

典型问题：判断 t 是否能被 $\sum_{i=1}^{m} a_i x_i$ 表示。由于 x 可被表示意味着 $x + x_1$ 也能被表示，于是可以按照 $x \bmod x_1$ 进行分类，x 向 $(x + a_j) \bmod a_1$ 连边。

我们只需知道某个余数最早可达时间即可。跑最短路即可，复杂度 $O(a_1 m \log a_1)$。

同余最短路

仔细思考，我们真的需要跑最短路吗？

可以发现这张图上最短路有个性质：边的顺序是任意的。具体而言，$6 = 1 + 2 + 2 + 1$ 和 $6 = 1 + 1 + 2 + 2$ 都是合法的最短路。

这启示我们直接以边权为阶段进行 DP。设 $f_{i,j}$ 表示只考虑前 i 种边权的最短路。考虑转移，类似完全背包，方程为 $f_{i,j} = \min(f_{i-1,j}, f_{i,(j-e_i) \bmod a_1} + k_i)$。

发现就是在 $\gcd(a_1, e_i)$ 个环上转移。断环为链后倍长，转移是简单的。总复杂度为 $O(a_1 m)$。

这个做法不仅时间复杂度更优，同时更易扩展。

Unknown Source

题目描述

给定长为 n 的字符串 S。你可以将若干个 S 拼接起来，拼接可以重叠，但要求重叠的部分必须相同。
求能得到的长度不超过 m 的字符串的长度种类数。
$n \le 10^6, m \le 10^{18}$。

Hint：字符串的 border 可划分为 $\log n$ 个等差数列。

Unknown Source

题目描述

给定长为 n 的字符串 S。你可以将若干个 S 拼接起来，拼接可以重叠，但要求重叠的部分必须相同。
求能得到的长度不超过 m 的字符串的长度种类数。
$n \le 10^6, m \le 10^{18}$。

Hint：字符串的 border 可划分为 $\log n$ 个等差数列。

Sol

朴素算法即为求出 S 的所有 border，则问题转化为令 $a_i = |S| - b_i$，b_i 为 border 长度，对这些 a_i 跑同余最短路，复杂度 $O(n^2 \log n)$。

根据结论，字符串的 border 可划分为 $\log n$ 个等差数列。设每个等差数列表示为 (a_i, d_i, k_i) 的形式，分别表示首项、公差、项数。

以等差数列为阶段 DP，在前述做法的基础上，发现就是加入了边权 d_i 最多用 k_i 次的限制。转移变成 $f_{i,j} = \min(f_{i-1,j}, f_{i,(j-a_i-td_i) \bmod a_1} + a_i + td_i), t \le k_i$。类似多重背包，按 $\bmod d_i$ 分类，断环为链后倍长，使用单调队列优化即可。

这部分内容相当丰富，甚至可以单开一课。

因此只选取部分较为有用的模型与技巧讲。

树链剖分

常用于维护路径信息。

对于重链剖分，一个有趣的应用是维护修改复杂度与点的度数相关的信息。

题目描述

对一棵树，维护如下操作：

1. 将一个点的所有儿子权值加一个数。
2. 查询一条链权值和。

树链剖分

Sol

对所有儿子暴力修改不现实。能不能少修改一点？
一条链在重链剖分中有个特性：轻边不超过 $\log n$ 条。而周知一个点的重儿子是唯一的。
这启示我们可以只修改一个点的重儿子，然后在这个点上打标记表示对所有轻儿子加一个数。
对于询问，在原来方法基础上，跳轻边时额外加上父亲的标记即可。

树链剖分

另外一个是维护毛毛虫修改，即与一条链距离不超过 1 的点集有关的操作。

- 修改：把每条重链给提出来，从上到下标记一遍 dfn，然后从上到下，遍历所有结点的轻儿子并赋 dfn 数组，然后每个点开 st,ed 表示这个点轻儿子的时间戳范围；
- 查询链：查询所在重链的贡献，单独查询当前链顶；
- 查询子树：每个点再记一个区间表示它的子树除去它的儿子的 dfn 范围，则子树则为刚才记的区间和、根到链底的重链、链上节点的轻儿子。

此处应该有图。

树链剖分——维护子树信息

此外，长链剖分常用来优化状态与深度相关的 DP。具体而言，每次父亲继承长儿子的 DP 状态，然后把其他儿子的 DP 状态合并上去。

这样的复杂度为所有长链的长度之和，即 $O(n)$。

这有点类似 dsu on tree，只不过后者改成了继承重儿子的状态，DP 状态个数与子树大小有关。

长链剖分，dsu on tree 与线段树合并都是维护树上所有子树信息的有力工具。

树链剖分——维护子树信息

此外，长链剖分常用来优化状态与深度相关的 DP。具体而言，每次父亲继承长儿子的 DP 状态，然后把其他儿子的 DP 状态合并上去。

这样的复杂度为所有长链的长度之和，即 $O(n)$。

这有点类似 dsu on tree，只不过后者改成了继承重儿子的状态，DP 状态个数与子树大小有关。

长链剖分，dsu on tree 与线段树合并都是维护树上所有子树信息的有力工具。

树链剖分

如果要动态改变树结构，大概率需要 LCT，但仍然是换汤不换药。
LCT 本质即为动态的树链剖分。由于超纲了，所以不打算展开来讲。

树链剖分

来点有趣的...

链剖分常应用于交互题中，特别是有关还原树的结构的题目。另外一种常用方法是剥叶子。

具体实现可看例题。

树链剖分

来点有趣的...

链剖分常应用于交互题中，特别是有关还原树的结构的题目。另外一种常用方法是剥叶子。

具体实现可看例题。

Unknown Source

题目描述

有一棵 n 个点的树，每次询问，你可以指定点 x 与点集 S，回答为 $\bigoplus_{y \in S} f(x,y)$，其中 $f(x,y)$ 为 x 到 y 路径上的点的标号和。
要求还原这棵树。
$n \leq 3000$，询问限制 $= 40000$。

Sol

先实现一点基本操作：记 d_u 为 u 到根的路径异或和（需要预处理），$f(u,v)$ 为 u,v 路径异或和。不难发现 $\mathrm{LCA}(u,v)=d_u\oplus d_v\oplus f(u,v)$。

考虑增量构造。对已知的点建虚树，并进行重链剖分。设当前在 x 点，要加入 u，且 u 一定在 x 子树中。

先判断 u 是否在重链上。若是，二分即可确定其位置，递归即可。

否则需要确定在哪个轻儿子中。不妨设轻儿子数量为偶数（目的是让 u 到 x 的路径抵消），令 v' 为查询 u 到轻儿子的结果，$s=\sum_{v\text{为轻儿子}} d_u\oplus d_v$，则有：

- $v'\oplus s$ 为轻儿子：递归至轻儿子即可。
- $v'\oplus s=0$：则 u 为 x 儿子。
- 其他情况：设 $w=v'\oplus s$，w 为 x 儿子且下面挂着 u 和 x 的另一个轻儿子。二分出另一个轻儿子即可。

如果考虑根到 u 的路径，只有 $\log n$ 条为轻边并需要执行上述操作，总询问次数为 $O(n\log n)$。

路径统计

常用：树分治（点分治、边分治）。

扩展：动态点分治（点分树）需要修改树的形态怎么办（如动态加叶子）？

定期重构。

路径统计

常用：树分治（点分治、边分治）。

扩展：动态点分治（点分树）需要修改树的形态怎么办（如动态加叶子）？

定期重构。

Unknown Source

并不是所有路径统计都是以上几种。有时候可以通过简单的拆贡献方法解决。

题目描述

有一棵 n 个点的树，有 m 条路径，每条路径有权值。q 次询问，每次询问与某条路径相交的路径权值之和。
$n, m, q \leq 3 \times 10^5$。

Sol

一条链与另一条链的交要么是一条链，要么是空集。

现在我们的目的是一条链的时候权值记为 x，否则为 0。

注意到交集的点数 $-$ 边数可以区分有没有交。于是对每条路径，将每个点权值 $+x$，每条边权值 $-x$ 即可。

虚树

关键：虚树的规模与关键点呈线性关系（$\leq 2k$）。
由此特性，虚树常常与数据结构配合，如线段树、分块等。

CF966E May Holidays

题目描述

有一棵树，每个节点有权值 a_i 。现在有 m 次操作，每次操作激活或禁用一个节点，每次操作后统计子树中禁用个数 $> a_i$ 且自身处于激活状态的节点个数。$n, m \leq 10^5$。

Sol

对操作序列分块。设块内操作涉及到的点为 $v_1, \ldots, v_c$，称其为关键点，则只有子树内有关键点的点有影响。

对关键点建虚树，虚树上每条边代表原树一条链。考虑每条边上的点按照子树中禁用的点数量 $-a_i$ 分组、排序，并维护一个指针代表第一个 > 0 的位置。

每次修改关键点，显然只要修改 $O(c)$ 条边的指针，移动量均为 $O(1)$，故修改的复杂度为 $O(c)$。每块操作开始前都要遍历整棵树，复杂度 $O(n)$。

取 $c = \sqrt{n}$，总复杂度为 $O(q\sqrt{n})$。

JOISC 2022 Sprinkler

杂题，不难但有点意思的题。

题目描述

有一棵 n 个点的树，每个点有权值。
q 次操作，一种是将距离一个点不超过 d 的点的权值在 $\bmod M$ 意义下乘以一个数，另一种为查询单点权值。
$n, q \leq 2 \times 10^5, d \leq 40$。

Sol

如果直接暴力跳父亲修改，发现我们需要解决“对某个点以外的所有儿子 $\times k$”这样的操作，但由于 M 不一定为质数，不能乘逆元。

这启示我们需要找到把“距离一个点不超过 k 的点”表示的集合刻画为**若干集合的不交并**。一个构造是拆成该点子树中与该点距离为 k 或 $k-1$ 的点集并上与他父亲距离 $\leq k-1$ 的点集（递归定义）。

然后就是简单的了。

JOISC 2023 Belt Conveyor

趣题一则。

题目描述

这是交互题。
给定一棵树，树的边有方向但未知。
每次询问，你可以指定若干条边和点，反转边的方向，并在每个点上放置一个物品。随后，物品将随机选择从所在位置为起点的边移动一步（不存在则不动），返回最后每个点上是否有物品。询问完毕后，所有边的方向将被还原。
请还原最初树边的方向。$N \leq 10^5$，操作次数限制为 30。

Sol

有意思的交互题。

将点按照深度分类。如果我们只在深度 d 的点上放物品，分类讨论：

- 有物体还在原处：确定了与它相连的所有边方向；
- 深度 $d+1$ 出现了物体：确定了这个位置连向父亲的边方向；
- 深度 d 的点有物品消失了：确定了这个位置连向父亲的边方向。

无论如何，其都可以确定至少深度 d 的点数条边的方向。但如果枚举深度逐个确定，还是太慢。

发现操作可以并行进行，只要 d 之间相差 ≥ 3 就互不影响。于是把原树按深度 $\bmod 3$ 分类，每次选择邻边有未知边的点最多的一类进行如上操作，这些点每个都能至少确定一条邻边方向。当然，需要排除已知边的干扰，直接将方向设为指向选择的点的方向即可。

每次至少知道 $1/3$ 的未知边的方向，询问次数 $\log_{1.5} n$，可以通过。

网络流

博大精深的板块。

最重要的点仍然是图论建模。

前置基础

- 网络流图的基本性质
- 最大流/费用流
- 上下界网络流

一些性质

- 流量守恒
- 最大流 $=$ 最小割
- EK 求费用流，每次增广单位流量的费用单调不减

一些扩展

- 上下界最大流/最小流/费用流怎么求？
- 费用流的图连出了负环怎么求？

对于后者，强制满流后变成反向的正权边，流量平衡类似上下界网络流处理即可。

一些扩展

- 上下界最大流/最小流/费用流怎么求？
- 费用流的图连出了负环怎么求？

对于后者，强制满流后变成反向的正权边，流量平衡类似上下界网络流处理即可。

常见建模

网络流中，考虑问题的角度一般有两种：最大流与最小割。

基础手法：

- 拆点
- INF 边防止被割
- ...

常见建模——集合划分

一般为将点划分为两个集合。

另外可能还有若干关系，形如集合 S 中的点在/不在同一集合会带来相应收益/损失。

从答案的形成方式考虑，建图方式有两种方式：答案取最大流，或是所有边权和 $-$ 最小割。

可以多个角度都考虑一下。

常见建模——集合划分（01 变量模型）

回忆一下：

- 每个变量取 0/1 值都会付出一定代价
- 两变量不等时有代价
- 多变量都取 0/1 时有收益
- ...

最后需要留意出现两侧的边都被割的状态。

THUPC 2022 分组作业

题目描述

有 $2n$ 个人，分为了 n 组，每组两个人。
首先，每个人可以表决愿意或不愿意与队友合作，带来的不满值分别为 c_i, d_i。如果两个人表决结果不同，会带来额外的不满值 e_i。如果一组中两个人都选择愿意，他们可以选择合作或不合作。
此外，还有 m 个单向关系 (x_i, y_i)。如果 x_i 没有合作且 y_i 选择愿意，带来 a_i 不满值；如果 x_i 选择不愿意但 y_i 成功合作，带来 b_i 不满值。
最小化不满值之和。$n \leq 5000, m \leq 10000$。

Sol

有点复杂的 01 变量模型。

首先是愿意或不愿意。按照套路，连接 $(S, x_i), (x_i, T)$ ，最小割割完后 x_i 属于 S 所在集合就是愿意，否则不愿意。此外，连 $(x_i, y_i, e_i), (y_i, x_i, e_i)$ 代表意见不一的代价。

合作需要双方都愿意，则 $(x_i, T), (y_i, T)$ 都会被割。建新点 z_i 代表合作，连 $(z_i, x_i, \text{INF}), (z_i, y_i, \text{INF})$ 即可。$(x_i, T), (y_i, T)$ 两者之一没有被割，则 S 到 z_i 必没有流量**（注意这里并不是直接建 (S, z_i) 的边）**。所以我们使用 S 到 z_i 有无流量代表有没有被割。

x_i 没有合作且 y_j 选择愿意会带来不满，相当于 S 到 z_i 无流量而 (S, y_j) 没有被割，故连 (y_j, z_i, a_i)；x_i 选择不愿意但 y_j 成功合作，相当于 S 到 z_j 有流量而 (x_i, T) 没有被割，连 (z_j, x_i, b_i) 即可。

常见建模——最大权闭合子图

点之间有依赖关系，且选择点的代价/收益不重复计算。

正权连源点，负权连汇点，依赖关系连 INF 边。

结果为正权点的和 − 最小割。

输出方案：与源点连通的点为选择的点。

Unknown Source

题目描述

有 n 个物品，每个物品有属性 a_i。有 m 条关系，形如选物品 i 就必须选 j。你需要选一个物品集合 S，最大化 $\frac{(\sum_{i\in S} a_i)^2}{\sum_{i\in S} a_i^2}$。

Sol

首先肯定二分。设限制为 x （显然 $x > 1$），则相当于最大化 $(\sum_{i\in S} a_i)^2 - x\sum_{i\in S} a_i^2$。

a_i^2 项好做。a_ia_j 项怎么办？考虑建 n^2 个新点 b_{ij}，收益为 a_ia_j，选了 b_{ij} 必须选 i, j 点即可。

由于 i 点的权值为 $(1-x)a_i^2 < 0$，因此不会出现选了 i, j 而不选 b_{ij} 的情况。

常见建模——网格图

常见的技巧为染色。本质是对图分层，注意多尝试几种染色方法，不一定都是黑白染色。

Trick：涉及网格图中的连通块问题，由于网格图也是平面图，因此连通块的边界可以转化为对偶图上的一个割集，从而套用网络流的模型。

常见建模——网格图

常见的技巧为染色。本质是对图分层，注意多尝试几种染色方法，不一定都是黑白染色。

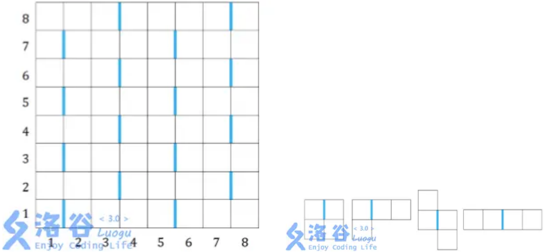
Trick：涉及网格图中的连通块问题，由于网格图也是平面图，因此连通块的边界可以转化为对偶图上的一个割集，从而套用网络流的模型。

CQOI 2017 老 C 的方块

经典题，较为复杂的染色与网络流建模。

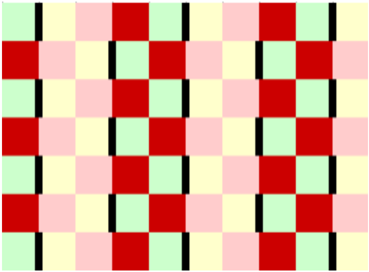
题目描述

下图中有若干黑色方块，每个方块移除有 w_i 的代价。求最小代价，使得黑色方块不包含以下图形（及其旋转、翻转）。



Sol

一张图解决问题。



常见建模——环（链）覆盖

链覆盖应该都知道怎么做吧。环覆盖呢？

题目描述

给定有向图，需要找若干不相交的环覆盖住顶点。
最大化覆盖到的点的字典序。
$n, m \le 3000$。

Sol

类似链覆盖，我们还是让出度与入度匹配。为了限定为环，我们强制最后必须为完美匹配。有些点没有被选上怎么办？连自环即可。于是我们得到了一个合法的环覆盖。

要最大化字典序，先固定前缀，需要求下一位最大是多少。

先用上下界强制前缀不能连自环（即被环覆盖），跑一遍最大流。然后从小到大扫连自环的点，在不经过前面的自环边的条件下，尝试对自环边退流即可。

常见建模——流量守恒/不等关系

利用流量守恒的恒等式，结合边的流量的上下界的限制的不等式。

本质上是解决一类特殊的线性规划问题。也可以考虑把题目抽象成线性规划问题（或其对偶），再转化为网络流。

似乎与区间覆盖/链覆盖关系紧密？

Unknown Source

题目描述

有 n 颗珍珠，每颗为蓝色与红色之一，满足如下要求：
有 3 种操作：

1. 将 x 号与 y 号珍珠所在箱子合并；
2. 删除 x 号珍珠；
3. x 号珍珠所在箱子有 $[l, r]$ 颗蓝珍珠。

试确定初始珍珠颜色，满足所有操作 3 的要求。
$n \leq 1000$。

Sol

首先按照操作建出并查集的操作树。没有删除操作时，操作树中儿子向父亲连有上下界的边即可。

考虑有删除操作，相当于要对 x 号珍珠所在箱子的点 y 强制分流。考虑流量守恒，只要 y 额外接一条流量恒为 1 的边出来连到 T 即可，这样流向父亲的流量就被分流了。

跑上下界可行流即可。

NOI 2008 志愿者招募

题目描述

给定一个项目需要 n 天完成，其中第 i 天至少需要 a_i 个志愿者。现有 m 类志愿者可供招募，第 j 类志愿者的工作时间为 $[s_j, t_j]$，招募费用为每人 c_j 元。目标是选择若干志愿者，满足每天的需求，并最小化总招募费用。

Sol

经典题。下面是一种比较机械化的方法。

设第 i 类志愿者选 t_i 个。为了叙述方便，下面是一个实例，其中有 3 类志愿者，分别覆盖 $[1,2],[2,3],[2,4]$。

列出所有不等式，形如：

$$
\begin{aligned}
t_1 &\geq a_1 \\
t_1 + t_2 + t_3 &\geq a_2 \\
t_2 + t_3 &\geq a_3 \\
t_3 &\geq a_4
\end{aligned}
$$

$$\text{minimize } t_1c_1 + t_2c_2 + t_3c_3$$

Sol

我们需要流量守恒，考虑额外设元 $k_1, k_2, k_3, k_4 \geq 0$，则有：

$$\begin{aligned} t_1 &= a_1 + k_1 \\ t_1 + t_2 + t_3 &= a_2 + k_2 \\ t_2 + t_3 &= a_3 + k_3 \\ t_3 &= a_4 + k_4 \\ \text{minimize } & t_1 c_1 + t_2 c_2 + t_3 c_3 \end{aligned}$$

发现 t_1, t_2, t_3 出现了多次，而周知流量并不能复用，怎么办？联想到每个志愿者工作时间为区间，考虑差分：

$$\begin{aligned} t_1 &= a_1 + k_1 \\ t_2 + t_3 &= a_2 + k_2 - a_1 - k_1 \\ -t_1 &= a_3 + k_3 - a_2 - k_2 \\ -t_2 &= a_4 + k_4 - a_3 - k_3 \\ -t_3 &= -a_4 - k_4 \\ \text{minimize } & t_1 c_1 + t_2 c_2 + t_3 c_3 \end{aligned}$$

Sol

我们需要流量守恒，考虑额外设元 $k_1, k_2, k_3, k_4 \geq 0$，则有：

$$\begin{aligned} t_1 &= a_1 + k_1 \\ t_1 + t_2 + t_3 &= a_2 + k_2 \\ t_2 + t_3 &= a_3 + k_3 \\ t_3 &= a_4 + k_4 \\ \text{minimize } & t_1 c_1 + t_2 c_2 + t_3 c_3 \end{aligned}$$

发现 t_1, t_2, t_3 出现了多次，而周知流量并不能复用，怎么办？联想到每个志愿者工作时间为区间，考虑差分：

$$\begin{aligned} t_1 &= a_1 + k_1 \\ t_2 + t_3 &= a_2 + k_2 - a_1 - k_1 \\ -t_1 &= a_3 + k_3 - a_2 - k_2 \\ -t_2 &= a_4 + k_4 - a_3 - k_3 \\ -t_3 &= -a_4 - k_4 \\ \text{minimize } & t_1 c_1 + t_2 c_2 + t_3 c_3 \end{aligned}$$

Sol

建图已经呼之欲出了：t_i 就是从 l_i 到 r_i+1 的流量，费用为 c_i。等式右侧，i 向 $i+1$ 连下界为 a_i 的边即可。

跑最小费用可行流即可。

常见建模——线性规划

前面提到，网络流本质上是解决一类特殊的线性规划问题。

有时候可以将题目抽象成线性规划问题（或其对偶），并进一步转化为网络流模型。

这个过程通常较为机械化，通用性较强。

线性规划对偶

首先介绍线性规划对偶（摘自 Wikipedia）：

问题甲	问题乙
最大化目标函数 $\max_x c^T x = \max_x \sum_i c_i x_i$	最小化目标函数 $\min_y b^T y = \min_y \sum_i b_i y_i$
n个变量$x_1, \ldots, x_n$ • $x_i \geq 0$ • $x_j \leq 0$ • $x_k \in \mathbb{R}$	n个限制条件$A^T y \lesseqqgtr c$ • 第i个限制条件为$\geq c_i$ • 第j个限制条件为$\leq c_j$ • 第k个限制条件为$= c_k$
m个限制条件$Ax \lesseqqgtr b$ • 第i个限制条件为$\geq b_i$ • 第j个限制条件为$\leq b_j$ • 第k个限制条件为$= b_k$	m个变量$y_1, \ldots, y_m$ • $y_i \leq 0$ • $y_j \geq 0$ • $y_k \in \mathbb{R}$

线性规划对偶

一个例子（摘自 Wikipedia）：

问题甲	问题乙
$\begin{aligned} \text{maximize } & 3x_1 + 4x_2 \\ \text{s.t. } & 5x_1 + 6x_2 = 7 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$	$\begin{aligned} \text{minimize } & 7y_1 \\ \text{s.t. } & 5y_1 \geq 3 \\ & 6y_1 \geq 4 \\ & y_1 \in \mathbb{R} \end{aligned}$

看似复杂，利用拼凑的思想，其实有一个比较自然的理解方式。

实际上，最大流与最小割表示成线性规划的形式也是对偶问题。

CF1307G Cow and Exercise

题目描述

给出一个 n 个点 m 条边的有向图，每条边有边权 w_i。
有 Q 次询问，每次询问给出一个 x 。你可以把一条边修改成 $w_i + x_i$，且 $x_i \geq 0, \sum x_i \leq x$。
你要通过修改边权使得从 1 到 n 的最短路径尽可能长，每次询问之间独立。
$n \leq 50, Q \leq 10^5$。

Sol

这道题有更简洁的理解方法，但这里想讲更通用的线性规划法。

设 $w_{u,v}$ 为边权，$x_{u,v}$ 为加的边权，d_u 为最短路，则容易列出不等式：

$$\begin{aligned} d_v &\le d_u + (w_{u,v} + x_{u,v}) \\ x_{u,v} &\ge 0 \\ \sum x_{u,v} &\le X \\ \text{maximize } d_n - d_1 \end{aligned}$$

Sol

这与费用流形式相差甚远。考虑对偶形式：

$$\begin{aligned}\sum_v c_{u,v} - c_{v,u} =& \begin{cases} -1 & u = n \\ 1 & u = 1 \\ 0 & \text{otherwise} \end{cases} \\ c_{u,v} + d_{u,v} - t =& 0 \\ c_{u,v}, d_{u,v}, t \geq& 0 \\ \text{minimize } tX + \sum c_{u,v} w_{u,v} \end{aligned}$$

第一项就是我们希望的流量守恒的式子（显然可以看出 1 为源点，n 为汇点，流量为 1），第二项相当于所有边流量限制为 t；而目标函数的求和部分就是费用流的总费用。

Sol

这与费用流形式相差甚远。考虑对偶形式：

$$
\begin{aligned}
\sum_v c_{u,v} - c_{v,u} = &\begin{cases} -1 & u = n \\ 1 & u = 1 \\ 0 & \text{otherwise} \end{cases} \\
c_{u,v} + d_{u,v} - t =& 0 \\
c_{u,v}, d_{u,v}, t \geq & 0 \\
& \text{minimize } tX + \sum c_{u,v} w_{u,v}
\end{aligned}
$$

第一项就是我们希望的流量守恒的式子（显然可以看出 1 为源点，n 为汇点，流量为 1），第二项相当于所有边流量限制为 t；而目标函数的求和部分就是费用流的总费用。

Sol

为了固定每条边的流量限制，整体乘 t^{-1}，有：

$$\sum_v c_{u,v} - c_{v,u} = \begin{cases} -t^{-1} & u = n \\ t^{-1} & u = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$0 \le c_{u,v} \le 1$$

$$\text{minimize } \frac{X + \sum c_{u,v} w_{u,v}}{t^{-1}}$$

令 $f = t^{-1}$，由 t 的任意性知 f 的任意性，从而

$$\sum_v c_{u,v} - c_{v,u} = \begin{cases} -f & u = n \\ f & u = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$0 \le c_{u,v} \le 1$$

$$\text{minimize } \frac{X + \sum c_{u,v} w_{u,v}}{f}$$

Sol

相当于在费用流图连 $(u, v, w_{u,v}, 1)$ 的边，要求最小化 $\frac{X+c}{f}$，c 为总费用。

显然，我们需要建出 $y = \frac{1}{f}x + \frac{c}{f}$ 的凸壳。用 EK 求费用流的时候，f 单调递增，$\frac{c}{f}$ 单调不减，正好符合要求。

凸费用

费用流的简单扩展。

具体而言，若一条边的费用关于流量为凸函数而非线性关系，则可以通过拆边的方式满足要求。

杂题

一些无法归类的题目。

POJ1147 PIGS

题目描述

有 n 个猪圈，第 i 个猪圈初始有 a_i 头猪。有 m 个顾客，每个顾客会指定几个猪圈，你可以先任意重分配这些猪圈中的猪，再卖给这个顾客最多 b_i 头猪。最大化卖猪的总数量。$n, m \leq 1000$。

Sol

难点在于处理完某个客户需求时，可以通过分配使得多出来的猪可以分配给后面的客户。

发现如果两个客户指定的猪圈有交，那么前面的客户可以通过把若干猪分配到猪圈的交上从而实现转移任意猪给后面的客户。

建图是简单的。

网络流相关构造

构造的方法极为多样，基本只能视具体题目而定。

一个常见的手法是：先忽略某些限制得出一组解，然后调整得到合法解："先构造后调整"。

ICPC 2017 WF Son of Pipe Stream

题目描述

一张 n 个点的图，1,2 号点分别为 W,F 液体的源点。两种液体分别遵循流量守恒。一条边的容量限制表现为 $w_i + vf_i \le c_i$，w_i, f_i, c_i 为这条边的 W 液体流量、F 液体流量与总容量。一条边中两种液体的流向必须相同。
设 F, W 为 F,W 液体的总流量。最大化 $F^a \times W^{1-a}$。
$n \le 300$。

Sol

注意到 v 没有用，以后直接设 $v=1$。设总的最大流为 A，F,W 最大值分别为 F_1,W_1，显然最后 $F\in[A-W_1,F_1]$。利用高中导数知识，不难求出目标函数最大时 F 的取值。

下面用构造证明这样的 F 一定能取到。

建超级源点 S，向 $1,2$ 分别连容量为 $F,A-F$ 的边。先直接跑出最大流为 A 的情况。

然后以正向边流量为容量建新图，从 1 出发跑最大流为 F 的情况。则每条边流 F 液体流量为新图流量，容量 - 新图流量为 W 液体流量。

显然这是可以构造出来的，由流量守恒也不难知构造合法。

Sol

注意到 v 没有用，以后直接设 $v=1$。设总的最大流为 A，F,W 最大值分别为 F_1,W_1，显然最后 $F\in[A-W_1,F_1]$。利用高中导数知识，不难求出目标函数最大时 F 的取值。

下面用构造证明这样的 F 一定能取到。

建超级源点 S，向 $1,2$ 分别连容量为 $F,A-F$ 的边。先直接跑出最大流为 A 的情况。

然后以正向边流量为容量建新图，从 1 出发跑最大流为 F 的情况。则每条边流 F 液体流量为新图流量，容量 - 新图流量为 W 液体流量。

显然这是可以构造出来的，由流量守恒也不难知构造合法。

联合省选 2022 学术社区

尝试不用欧拉回路。

题目描述

有 n 个人和 m 个消息，一个消息有三种情况：

- `A B loushang`：发消息的人叫 A，它产生贡献当且仅当下一条消息是 B 发的；
- `A B louxia`：发消息的人叫 A，它产生贡献当且仅当上一条消息是 B 发的；
- 其他情况：A 发的一条学术消息，永远不产生贡献。

每人至少有一条学术消息，你要找到一种重排消息的方案，使得所有消息贡献之和最大。
$m \le 2.5 \times 10^5$。

性质 C：若存在一条消息是 `A B loushang`，则不存在一条消息是 `B A louxia`。
难点在构造方案。

联合省选 2022 学术社区

尝试不用欧拉回路。

题目描述

有 n 个人和 m 个消息，一个消息有三种情况：

- `A B loushang`：发消息的人叫 A，它产生贡献当且仅当下一条消息是 B 发的；
- `A B louxia`：发消息的人叫 A，它产生贡献当且仅当上一条消息是 B 发的；
- 其他情况：A 发的一条学术消息，永远不产生贡献。

每人至少有一条学术消息，你要找到一种重排消息的方案，使得所有消息贡献之和最大。

$m \leq 2.5 \times 10^5$。

性质 C：若存在一条消息是 `A B loushang`，则不存在一条消息是 `B A louxia`。

难点在构造方案。

Sol

首先考虑有 C 性质怎么做。

考虑每句话接下来可以接哪些话，不难发现：

- 对于形如 `A B loushang` 的话，其后只能接发出者为 B 的话
- 对于形如 `A B louxia` 的话，其前只能接发出者为 B 的话

于是可以建图，第 i 句话连向所有可以接到其后面的话，被所有可以接到其前面的话连。

于是就有个网络流思路了：

- 考虑将第 i 句话拆成出度与入度 2 个点，称为 out_i, in_i，然后对于 n 个人中每个人都建 C_j, D_j 两个虚点；
- 对于第 i 句话，设发出者为 A，连接 $(S, out_i, 1), (in_i, T, 1), (D_A, in_i, 1), (out_i, C_A, 1)$；
- 如果形如 `A B loushang`，再连接 $(out_i, D_B, 1)$；
- 如果形如 `A B louxia`，再连接 $(C_B, in_i, 1)$。

Sol

不难发现，这是一个类似二分图匹配的东西（只是优化了一下连边），当 out_i 有流量到 in_j 时，即表示最终方案中第 i 句话后接第 j 句话。

直接跑网络流，可以发现第一问的答案均正确，这给了我们信心。接下来考虑如何构造方案。

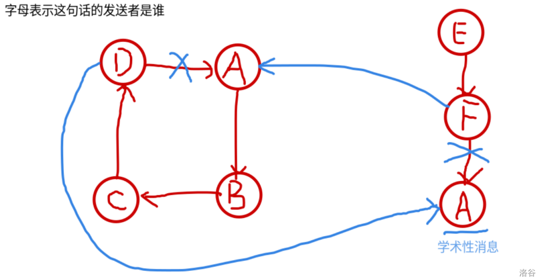
不难通过残量网络的情况来推出每个点匹配了哪个点，进而可以知道每句话的下一句话是哪句话，于是可以把每句话向其下一句话连边。

Sol

然而，这显然可能不合法，原因是可能有环，可能出现 out_1 匹配 in_2，out_2 匹配 in_3，out_3 匹配 in_1 这样的情况。

不过，这样得到的图显然由若干条链 + 若干个环组成，并且环上的点的类型均相同（都是楼上型或都是楼下型）。

考虑将环拆开，由于题目中保证每个人至少都有一条学术型消息，所以可以把环接到学术性消息去，以楼上型的环为例，如图：



Sol

这样就可以把环接到学术性消息去，可以方便地用双向链表维护。楼下型的环同理。最后输出每条链即可。

至于没有 C 性质，不难发现把这样的两句话缩起来一定不劣，然后就变成类似学术的东西了。套用之前做法即可。

杂项——平面图最小割转最短路

虽然网络流的效率较高，但面对很大规模的数据（10^6）还是力不从心。
如果是平面图，可以发现最小割等同于对偶图上的最短路。

杂项——平面图最小割转最短路

大多数时候是直接构造，不过也有对连通平面图的通用方法——最小左转法。

步骤如下：

1. 把所有无向边拆成正反两条有向边；
2. 把所有有向边存在起点的 vector 中，所有 vector 中向量逆时针极角排序。
3. 从一条有向边出发，设当前为 (x, y)，在 y 的 vector 中找到 (y, x)，转移到其上一条有向边。如此循环直至回到原点，这样就形成了一个面。找出所有面即可。

杂项——平面图最小割转最短路

大多数时候是直接构造，不过也有对连通平面图的通用方法——最小左转法。

步骤如下：

1. 把所有无向边拆成正反两条有向边；
2. 把所有有向边存在起点的 `vector` 中，所有 `vector` 中向量逆时针极角排序。
3. 从一条有向边出发，设当前为 (x, y)，在 y 的 `vector` 中找到 (y, x)，转移到其上一条有向边。如此循环直至回到原点，这样就形成了一个面。找出所有面即可。

杂项——最小割的方案、必经边、可行边

- 方案构造：残量网络中与源点联通的点集视为一个连通块；
- 必经边：(u,v) 中 u 与 S、v 与 T 分别在同一个 SCC 中；
- 可行边：(u,v) 满流，且 u 与 v 不在同一 SCC 中。

证明考虑微调 (u,v) 的容量（$\pm\mathrm{eps}$），必经边相当于 (u,v) 增大则最小割增大，可行边相当于 (u,v) 减小则最小割减小。

杂项——最小割树（Gomory-Hu Tree）

定义

对于图 G，其上的最小割树是这样一棵树：对所有树边 (u, v, w)，满足 u, v 之间的在原图上的最小割为 w，且割后形成的两个连通块为树删去该边形成的两个连通块。

这种最小割树实现较为复杂。实践中，由于最小割树一般来求任意两点的最小割数值，故采取以下方法构造另一种没有那么强的最小割树：

1. 在当前连通块选取两点 s, t，求 s 至 t 的最小割；
2. 在树上加入边 (s, t, w)，w 为最小割容量；
3. 删除割集，递归进入两个连通块，重复上述过程。

这样求出来的最小割树上，任意两点最小割为路径上边权最小值。

杂项——最小割树（Gomory-Hu Tree）

定义

对于图 G，其上的最小割树是这样一棵树：对所有树边 (u, v, w)，满足 u, v 之间的在原图上的最小割为 w，且割后形成的两个连通块为树删去该边形成的两个连通块。

这种最小割树实现较为复杂。实践中，由于最小割树一般来求任意两点的最小割数值，故采取以下方法构造另一种没有那么强的最小割树：

1. 在当前连通块选取两点 s, t，求 s 至 t 的最小割；
2. 在树上加入边 (s, t, w)，w 为最小割容量；
3. 删除割集，递归进入两个连通块，重复上述过程。

这样求出来的最小割树上，任意两点最小割为路径上边权最小值。

杂项——最小割树（Gomory-Hu Tree）

定义

对于图 G，其上的最小割树是这样一棵树：对所有树边 (u,v,w)，满足 u,v 之间的在原图上的最小割为 w，且割后形成的两个连通块为树删去该边形成的两个连通块。

这种最小割树实现较为复杂。实践中，由于最小割树一般来求任意两点的最小割数值，故采取以下方法构造另一种没有那么强的最小割树：

1. 在当前连通块选取两点 s,t，求 s 至 t 的最小割；
2. 在树上加入边 (s,t,w)，w 为最小割容量；
3. 删除割集，递归进入两个连通块，重复上述过程。

这样求出来的最小割树上，任意两点最小割为路径上边权最小值。

DFS 生成树

生成树将图的结构划分为了一棵树与非树边，而两者都有比任意图更丰富的性质。例如：

- 生成树上每个点有唯一父亲。
- 无向图上的 DFS 生成树不存在横叉边，即非树边都是连接祖先与子孙。
- ...

这对于某些构造题有奇效，特别是需要点权满足某些条件的题。

CF1218G Alpha planetary system

题目描述

给定无向连通三分图，定义每个点的权值为所有与它相连的边的权值之和。
需要对每一条边赋权 $1, 2, 3$，使得图中每条边所连接的两点权值不同。
求赋权方案。$n, m \leq 10^5$。

Sol

设 w_x 表示点 x 的权值 $\bmod 3$，c_x 表示 x 的所属集合编号（$c_x \in \{0,1,2\}$），v_i 表示第 i 条边的权值。

一个直接的想法是使所有 $w_x = c_x$，当然这不一定能做到，具体见下文。

首先，根据套路，随便建一颗生成树出来。设根为 rt。

先将所有非树边权值设为 3。对于点 u，若其子树内所有树边权值都确定了，则 u 到父亲的树边的权值可以唯一确定。

然而，rt 没有到父亲的边，此时要做出调整：

- 如果此时 $w_{rt} = c_{rt}$，则方案已经合法。
- 否则，我们需要找到包含 rt 的一个环进行调整。

Sol

提取出包含 rt 的一个环，设环上边的编号依次为 $e_1, e_2, \ldots, e_k$，其中 e_1, e_2 两条边均有一个端点是 rt。

偶环是没用的，因为在偶环上对权值调整只能形如 $v_{e_1}+1, v_{e_2}+2, v_{e_3}+1, v_{e_4}+2, \ldots$，无法改变任何一点的权值。

奇环是有用的，考虑如下调整：

- 当 $w_{rt}+1 \equiv c_{rt} \pmod 3$，执行 $v_{e_1}+2, v_{e_2}+2, v_{e_3}+1, v_{e_4}+2, v_{e_5}+1, \ldots$。显然环上除了 rt 其他点的权值均未改变。
- 当 $w_{rt}+2 \equiv c_{rt} \pmod 3$，执行 $v_{e_1}+1, v_{e_2}+1, v_{e_3}+2, v_{e_4}+1, v_{e_5}+2, \ldots$。

因此，当原图中存在奇环，将 rt 取奇环上的任意一点，一定可以构造出合法方案。

Sol

如果没有奇环呢？容易想到原图是二分图。

考虑重新将所有点划分为 2 个集合，这可以通过二分图染色实现。设 c_x 表示 x 新的所属集合编号（$c_x \in \{1,2\}$）。

不失一般性地设 $c_{rt}=1$，还是像 Part 1 那样将处理树边，最后考虑 rt 的情况：

1. $w_{rt} \neq 2$。此时方案已经合法。因为对于与 rt 之间有边的点 u 必有 $w_u = 2 \neq w_{rt}$。
2. rt 的度数 $=1$。此时方案已经合法。设与 rt 之间有边的点为 u，显然 u 的度数 >1，则 u 的权值必然 $>rt$ 的权值。
3. $w_{rt}=2$ 且 rt 的度数 >1。选出与 rt 之间有边的两个点 u,v，将 $(rt,u),(rt,v)$ 两条边的权值 $+1$ 即可。调整后 $w_{rt}=1, w_u=w_v=0$，可以证明满足要求。

本题似乎还可以基于生成树随机化乱搞，不过没有仔细了解过实现。

Sol

如果没有奇环呢？容易想到原图是二分图。

考虑重新将所有点划分为 2 个集合，这可以通过二分图染色实现。设 c_x 表示 x 新的所属集合编号（$c_x \in \{1, 2\}$）。

不失一般性地设 $c_{rt} = 1$，还是像 Part 1 那样将处理树边，最后考虑 rt 的情况：

1. $w_{rt} \neq 2$。此时方案已经合法。因为对于与 rt 之间有边的点 u 必有 $w_u = 2 \neq w_{rt}$。
2. rt 的度数 $= 1$。此时方案已经合法。设与 rt 之间有边的点为 u，显然 u 的度数 > 1，则 u 的权值必然 $> rt$ 的权值。
3. $w_{rt} = 2$ 且 rt 的度数 > 1。选出与 rt 之间有边的两个点 u, v，将 $(rt, u), (rt, v)$ 两条边的权值 $+1$ 即可。调整后 $w_{rt} = 1, w_u = w_v = 0$，可以证明满足要求。

本题似乎还可以基于生成树随机化乱搞，不过没有仔细了解过实现。

SCC

连通分量相关的题目，通常考虑分量内部两点关系对题意的影响，进而将没有特殊性质的图化为 DAG、树等，获得更多特殊性质解题。

SCC 性质：

- 使用 Tarjan 得出的 SCC，缩点后的拓扑序为标号逆序。
- 竞赛图缩点后的结果是一条链。

Unknown Source

有意思的构造题。

题目描述

给定有向图，求添加多少条边可以将其变为强连通图。
$n \leq 10^6$。

Sol

SCC 缩点后，答案的下界显然是 $\max($入度为 0 的点数, 出度为 0 的点数$)$。下面通过构造证明能够取到。

缩点后可能图不联通，先顺次连接使得图弱联通。显然我们只需要关注入度或出度为 0 的点的可达情况。

我们希望把所有入度或出度为 0 的点“串在一起”，因此希望找到入度为 0 的点到出度为 0 的点的匹配，这样可以串成环来互相可达。考虑若干条路径 (u, v)，满足：

- u 入度为 0，v 出度为 0；
- u, v 均两两不同；
- 对于入度为 0 的点 u'，一定存在某个 v 使得 u' 可到 v。
- 对于出度为 0 的点 v'，一定存在某个 u 使得 u 可到 v'。

构造是简单的：从每个入度为 0 的点出发 dfs，沿途打标记，如果到了出度为 0 的点就增加有一条路径，否则遇到打过标记的点就退出。可以证明这满足所有要求。

Sol

将所有路径顺次连接成环。而对于入度为 0 且不属于路径起点的点 u'，一定存在某个 v 使得 u' 可到 v，连接 $v \to u'$ 即可。对于出度为 0 同理。这样所有点都互相可达了。

DCC

v-DCC 重要性质：

- 圆方树还记得吧？
- v-DCC 内部任意两点所有简单路径的并等于 v-DCC。实际上，对于简单路径这类问题，由于其经过割点后不可返回，所以 v-DCC 是常用的工具。

e-DCC：内部任意两点都有至少两条边不重复的简单路径。

其实上述性质本质上都是最大流——最小割定理的特殊形式。

Menger's Theorem

最小割集的大小等于任意在所有顶点对之间可以找到的不相交路径的最大数量。
有边联通、点联通两个版本。

DCC

v-DCC 重要性质：

- 圆方树还记得吧？
- v-DCC 内部任意两点所有简单路径的并等于 v-DCC。实际上，对于简单路径这类问题，由于其经过割点后不可返回，所以 v-DCC 是常用的工具。

e-DCC：内部任意两点都有至少两条边不重复的简单路径。

其实上述性质本质上都是最大流——最小割定理的特殊形式。

Menger's Theorem

最小割集的大小等于任意在所有顶点对之间可以找到的不相交路径的最大数量。有边联通、点联通两个版本。

CF487E Tourists

题目描述

给定无向联通图，每个点有点权，维护两种操作：

- 修改点权；
- 询问两点所有简单路径中，能经过的最小点权。

$n, q \leq 2 \times 10^5$。

Hint：v-DCC 内部任意两点简单路径并为整个 v-DCC，可构建圆方树。

CF487E Tourists

题目描述

给定无向联通图，每个点有点权，维护两种操作：

- 修改点权；
- 询问两点所有简单路径中，能经过的最小点权。

$n, q \leq 2 \times 10^5$。

Hint：v-DCC 内部任意两点简单路径并为整个 v-DCC，可构建圆方树。

Sol

考虑 v-DCC。由于简单路径过了割点无法再返回，且 v-DCC 内部任意两点简单路径并为整个 v-DCC。

考虑建圆方树，设方点权值为周边圆点的权值最小值，则询问的答案即为路径最小值，树剖维护即可。

对于修改操作，修改圆点会影响周边所有的方点，不能直接修改。发现每个圆点只有唯一的父亲方点，我们修改方点权值为儿子圆点的权值最小值，这样修改就只用修改。

至于询问，若跳链最后到方点，多询问一次父亲圆点即可。

CF51F Caterpillar

不算困难的题。

题目描述

给定一张无向图，每次操作合并两个点并继承原来的边，求变成毛毛虫的最小操作次数。
毛毛虫：存在一条简单路径使得所有点到路径的距离不超过 1，且不存在环(可以有自环，不能有重边)。
$n \leq 2000, m \leq 100000$。

Sol

不能有环 $\iff$ 是森林 $\iff$ e-DCC 都缩成了单点。

首先肯定要通过合并操作把 e-DCC 缩点，随后问题转化为一棵树上选一条主链，最小化操作次数。

模拟合并的过程，可以发现其实就是不停地把与主链相连的、且不是叶子的点合并到主链上，故操作次数为 $n-$ 主链长度 $-$ 叶子个数，则选择树的直径即可。

最后把森林合并为一棵树即可。

一个小 trick

求 SCC 的算法除了 Tarjan 之外，还有 Kosaraju 算法，可自行了解。

由于其实现只用到了 2 次 DFS，故可以配合 bitset 做到单次 $O(n+\frac{m}{w})$ 的复杂度。

也许会有用吧。

广义串并联图

定义：不存在同胚于 K_4 的子图的图。如果联系到电阻网络之类很容易理解这一点。

性质：

- 是平面图；
- $m \le 2n$（无重边）；
- **（最重要）**可以通过以下操作缩为单点：删一度点，缩二度点，叠合重复边。

最后一种操作其实不仅限于广义串并联图，对一般图也适用。当 $m \le n + k$ 时，容易证明操作后 n, m 均为 $O(k)$ 级别，有时候可以大大减小图的规模。

JOI Open 2022 通学路

题目描述

给定带权无向图，询问从 1 到 N 是否存在长度大于最短路的简单路径。
$N, M \leq 2 \times 10^5$。

Hint 1：考虑整张图为点双。

Hint 2：考虑 $M - N$ 比较小的情况。

JOI Open 2022 通学路

题目描述

给定带权无向图，询问从 1 到 N 是否存在长度大于最短路的简单路径。
$N, M \le 2 \times 10^5$。

Hint 1：考虑整张图为点双。

Hint 2：考虑 $M - N$ 比较小的情况。

Sol

直接思考没有什么思路。根据 Hint 2，联想到广义串并联图的缩合操作。

- 删一度点：走过去就不能走回来了，直接删即可。
- 缩二度点：直接缩即可。
- 叠合重复边：两条边不一样就标记为 -1，否则直接合并。

缩合后直接搜索可以得到 $M-N$ 较小的部分分。

想象一下不存在题中路径的场景。先考虑 Hint 1 的情形，大胆猜想应该等价于与 $1,N$ 相连的边只有 $(1,N,*)$ 唯一一条且边权不为 -1。

一般情形也可以转化为 Hint 1：多连一条 $(1,N,dis_{1,N})$ 的边，然后只考虑 $1,N$ 所在的点双即可。

Sol

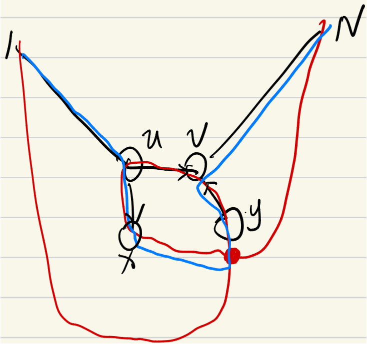
猜想的证明

需要证明如果不是这样的情形，那么一定存在符合条件的路径。
在缩合完毕后，显然所有点的度数 ≥ 3。
对每条边 (x, y, w)，一定有 $dis_{1,x} + w + dis_{y,N} = dis_{1,N}$ 或 $dis_{1,y} + w + dis_{x,N} = dis_{1,N}$。前者将 (x, y) 定向为 $x \to y$，后者为 $y \to x$。
标记入度大于出度的点为红，否则为蓝。考虑 1 到 N 的最长简单路径，1 和与 1 相邻的点为蓝，N 和与 N 相邻的点为红。
故必然存在相邻两点 u, v 使得 u 为蓝，v 为红。则 u 的出度 ≥ 2，v 入度 ≥ 2，存在 x, y 使得 $u \to x, y \to v, x \neq v, y \neq u$。

Sol

猜想的证明

由最长简单路径的性质，可以证明 $1 \to y \to v \to u \to x \to N$ 为简单路径 ($1 \to y, x \to N$ 走最短路)，见下图。这条路径显然比 $1 \to N$ 最短路长，得证。



建模

依然是老套路：观察性质，减少无用边数；或者与数据结构结合。

最小哈密顿生成树

题目内容

平面上 n 个点，两点间边权为哈密顿距离，求最小生成树。
$n \leq 2 \times 10^5$。

Sol

向八个方向最近的点连边即可。

CODE FESTIVAL 2016 Final Zigzag MST

题目内容

给定 N, Q，对一张 N 个点的图做 Q 次加边操作，每次给定 A_i, B_i, C_i，按顺序连边
$(A_i, B_i, C_i), (A_i + 1, B_i, C_i + 1), (A_i + 1, B_i + 1, C_i + 2), (A_i + 2, B_i + 1, C_i + 3)$ 等（编号对 N 取模）。求最小生成树。
$N, Q \leq 2 \times 10^5$。

Sol

考虑 Kruskal 算法的过程，我们在考虑 (A_i+1, B_i, C_i+1) 时，A_i 与 B_i 必然已经连通，这两个点之间没有区别，故可以改成 (A_i+1, A_i, C_i+1)。对后续的边也是同理，最后发现我们将 Z 字型路径转化为了一条弦 $+$ 两条边权为等差数列的链的结构。

显然，对于环上相邻两点之间的边，我们只需保留边权最小的那条。直接断环为链扫一遍即可。

Boruvka 算法

边数太多，并不一定是优化建图。

Boruvka 算法能够在不访问所有边的情况下，以 $O(n \log n)$ 计算最小生成树。通常这些边权按照一定规律生成。

具体而言，每一轮，每个连通块寻找向其他连通块的最小边权，合并所有连通块；由于每一轮连通块数量减少一半，故只会执行 $O(\log n)$ 次。

CF1550F Jumping Around

题目描述

数轴上顺次有 n 个点 $a_1 < a_2 < \cdots < a_n$。
有一只小青蛙，初始时在 a_s 处。小青蛙有两个参数：步长 d 和灵活程度 k。其中，步长 d 是确定的，而灵活程度 k 是可以调整的。
小青蛙可以从某个点跳到另一个点。但这是有要求的：小青蛙能从 a_i 跳到 a_j，当且仅当 $d - k \le |a_i - a_j| \le d + k$。
给定 $a_1, \ldots, a_n$ 和 d。你需要回答 q 次询问，每次询问给定一个下标 i 和灵活程度 k，回答此时的小青蛙能否跳到 a_i。
$n, q \le 2 \times 10^5$。

Sol

比较模板的题。

记 i 到 j 的边权为 $||a_i - a_j| - d|$，则询问相当于询问 s 能否经过不大于 d 的边到 i 点。

这是个经典问题，使用 Kruskal 重构树可以解决。现在的目标是建出该树，其实相当于建出最小生成树。

边有 $O(n^2)$ 条，但边权有规律。考虑 Boruvka 算法，找从某点出发边权最小的边，显然是查询距离 $a_i - d$ 和 $a_i + d$ 最近的点。

记得查询之前把所在联通块的点删除。

最小欧几里得生成树

题目内容

平面上 n 个点，两点间边权为欧氏距离，求最小生成树。
$n \leq 4 \times 10^4$。

Sol

正经的做法请自行学习 Voronoi 图。

下面是一个有些乱搞的做法，复杂度依赖于 k-D Tree 寻找最近点对的复杂度，因此没有保证。

考虑 Boruvka 算法，每次需要支持查询一个联通块内所有点到联通块外得最近点。在 k-D Tree 中把联通块内的点删去后查询最近点即可。

一共需要 $O(n \log n)$ 次平面最近点对查询，随机数据下表现较优秀。

LCT 求最小生成树

虽然标题很吓人，但实际上很多题并不需要 LCT。暴力维护也是可以的。

这种方法主要针对加边顺序任意的情况，通用性较强。核心思想很简单：每次加入边后删除环上边权最大的边。

用 LCT 固然可以快速实现上述操作，不过很多时候暴力 dfs 也足够了。

一些可能有用的性质

最小生成树的一些性质：

- 相同权值的边，加入边数一定相同；加入小于等于阈值的边，连通状况相同；
- 两个边集分别做最小生成树，把两棵生成树的边最后再做一次最小生成树，得到的是原边集的最小生成树（边集的可合并性）；
- 两个点集分别做最小生成树，则合并后只需考虑原来两棵最小生成树的边及两个点集之间的新边。此时可以考虑维护额外边涉及的点建虚树（点集的可合并性）。

证明利用 Kruskal 算法过程不难。

最后两个性质经常配合数据结构。

北京市赛 2025 最小生成树

题目描述

给定 n, k，求有多少种不同的 n 个点的无向连通图满足：

- 图中没有自环，且任意两个点之间至多有一条边。
- 所有边的边权为 $[1, k]$ 之间的整数。
- 对于图中的每条边，都至少存在一棵最小生成树包含该边。

对 998244353 取模。$n \le 5 \times 10^4, k \le 10$。

Sol

k 很小，依次考虑每种边权的边，假设当前 $\le i-1$ 的边都加入了，考虑哪些边边权可以是 i。

不难发现，只有两端在不在同一联通块的边边权才能是 i，这样的边数量只与加入 $\le i-1$ 的边后的连通情况有关。

于是枚举 n 所在连通块的大小 s，设 $f(i,n), g(i,n)$ 分别代表大小为 n 的连通块、任意图时的答案，转移是简单的：

$$g(i,n) = \sum_{s=1}^{n} \binom{n-1}{s-1} 2^{s(n-s)} f(i-1,s) g(i,n-s)$$

$$f(i,n) = g(i,n) - \sum_{s=1}^{n-1} \binom{n-1}{s-1} f(i,s) g(i,n-s)$$

使用分治 NTT / 多项式 $\ln / \exp$ 即可 $O(kn\log n)$ 或 $O(kn\log^2 n)$ 解决。

SDOI 2019 世界地图

题目描述

给定 $n \times m$ 列的带权网格图，第一列和最后一列相连。
有 q 组询问，每次会给出 l_i 和 r_i，询问禁用 $[l_i, r_i]$ 列的点后，剩下的点的最小生成树。
$n \leq 100, m \leq 10000, q \leq 10000$。

Sol

询问相当于合并前缀和后缀的最小生成树，再加入 n 条新边后询问最小生成树。

记这 n 条新边涉及的点为关键点，加入新边的过程相当于连新边后删除环上权值最大的边，相当于要支持查询关键点两两路径最大边权、断开边、连接边。

将关键点拿出来建虚树，边权为在原最小生成树路径上的最大边权，直接暴力维护即可。

至于预处理每一个前缀涉及关键点的虚树，可以把第 1 列和第 i 列的点设为关键点，用上面方法加入新点、边即可。

JOISC 2022 复兴计划

题目描述

给定连通无向图，有 q 次询问，每次指定 x，对每条边 i，将边权改为 $|w_i - x|$，求最小生成树。
$n \leq 500, m \leq 10^5, q \leq 10^5$。Bonus：$q \leq 10^6$。

Sol

一个暴力的想法是把边按 w_i 排序，预处理每个前缀的边权取负的最小生成树和每个后缀的最小生成树，每次询问直接合并即可。时间复杂度 $O(qn\alpha(n))$，表现已经相当优秀了。

一条边是否会被加入最小生成树，只与按边权排序后其前面的边有关。考虑 x 从小到大移动的过程，逐渐接近 w_i 的时候，$|w_i - x|$ 优先级逐渐超过一些边；而 x 越过 w_i 不断增大时，$|w_i - x|$ 优先级又被一些边超过。显然，w_i 被加入最小生成树，对应的 x 一定一段区间。

考虑如何求出这个区间。当 $x < w_i$ 时，$w_j > w_i$ 都还不需要考虑，故只考虑 $w_j < w_i$ 的边已经形成的最小生成树，如果 w_i 加入时没有成环，或是可以替代环上某条边，那么其就可以加入最小生成树。设环上边权最小为 w_k，则 $x > \frac{w_k + w_i}{2}$ 时 $|w_k - x| > |w_i - x|$，故可以确定区间左端点，同时也确定了 w_k 的右端点。随后，删除 w_k 边并加入 w_i 即可。

求出所有边的有效区间后，询问是简单的。

暴力维护或 LCT 维护，时间复杂度 $O(nm)$ 或 $O(m \log n)$。

Sol

一个暴力的想法是把边按 w_i 排序，预处理每个前缀的边权取负的最小生成树和每个后缀的最小生成树，每次询问直接合并即可。时间复杂度 $O(qn\alpha(n))$，表现已经相当优秀了。

一条边是否会被加入最小生成树，只与按边权排序后其前面的边有关。考虑 x 从小到大移动的过程，逐渐接近 w_i 的时候，$|w_i - x|$ 优先级逐渐超过一些边；而 x 越过 w_i 不断增大时，$|w_i - x|$ 优先级又被一些边超过。显然，w_i 被加入最小生成树，对应的 x 一定一段区间。

考虑如何求出这个区间。当 $x < w_i$ 时，$w_j > w_i$ 都还不需要考虑，故只考虑 $w_j < w_i$ 的边已经形成的最小生成树，如果 w_i 加入时没有成环，或是可以替代环上某条边，那么其就可以加入最小生成树。设环上边权最小为 w_k，则 $x > \frac{w_k+w_i}{2}$ 时 $|w_k - x| > |w_i - x|$，故可以确定区间左端点，同时也确定了 w_k 的右端点。随后，删除 w_k 边并加入 w_i 即可。

求出所有边的有效区间后，询问是简单的。

暴力维护或 LCT 维护，时间复杂度 $O(nm)$ 或 $O(m \log n)$。

欧拉回路

怎么求：应该不用多说。

k-链覆盖：奇度点两两配对连虚边。

混合图：

- 需要先确定每条无向边方向。
- 考虑**先构造再调整**。随便钦定方向，令 d_i 为出度与入度之差 $/2$，若不为整数说明度数为奇数，显然不存在欧拉回路。
- 然后考虑反转边。反转 $u \to v$ 相当于 d_u 给 d_v 1 的度数。这类似网络流的过程。
- 对 $d_u > 0$，需要给出 d_u 的度数，源点向其连容量为 d_u 的边；否则向汇点连边，容量为相反数。
- 如果满流则存在定向方案。构造是平凡的。

欧拉回路

怎么求：应该不用多说。

k-链覆盖：奇度点两两配对连虚边。

混合图：

- 需要先确定每条无向边方向。
- 考虑**先构造再调整**。随便钦定方向，令 d_i 为出度与入度之差 $/2$，若不为整数说明度数为奇数，显然不存在欧拉回路。
- 然后考虑反转边。反转 $u \to v$ 相当于 d_u 给 d_v 1 的度数。这类似网络流的过程。
- 对 $d_u > 0$，需要给出 d_u 的度数，源点向其连容量为 d_u 的边；否则向汇点连边，容量为相反数。
- 如果满流则存在定向方案。构造是平凡的。

哈密顿图

哈密顿图：含有哈密顿回路的图。

这类题目相对欧拉回路少见，难度也更大。一般情况下，在涉及哈密顿图的问题中，图有特殊性质。

典型：

- 如果任意两点度数和 $\geq n$，则图有哈密顿回路
- 最小度数 $\geq \frac{n}{2}$ 的图一定有哈密顿回路（上一结论的特殊情况）
- 竞赛图一定有哈密顿路径
- 特别的，强连通竞赛图一定有哈密顿回路

上述结论的证明都用到了增量构造法。这种方法在处理此类问题时非常重要。

哈密顿图

哈密顿图：含有哈密顿回路的图。

这类题目相对欧拉回路少见，难度也更大。一般情况下，在涉及哈密顿图的问题中，图有特殊性质。

典型：

- 如果任意两点度数和 $\geq n$，则图有哈密顿回路
- 最小度数 $\geq \frac{n}{2}$ 的图一定有哈密顿回路（上一结论的特殊情况）
- 竞赛图一定有哈密顿路径
- 特别的，强连通竞赛图一定有哈密顿回路

上述结论的证明都用到了增量构造法。这种方法在处理此类问题时非常重要。

Unknown Source

题目描述

给定 N 个点的完全图，每条边是红色或者蓝色。对于每个顶点 x，找出从 x 出发的一条尽量短的路径，经过每个节点，并且按顺序经过的边的颜色交替最多一次。

$N \leq 2000$。

Sol

答案下界显然为 N，大胆猜想下界能取到，即哈密顿路。

通过如下增量构造可以说明正确性：

- 设前 $N-1$ 个结点存在满足题意的路径。
- 如果路径所有边同色，直接把 N 放在最后即可。
- 否则，不妨设路径上在 x 处出现了颜色交替，设从红色变为蓝色，若 (x, N) 为红色，把 N 放在 x 后，否则放在 x 前。

不难验证这是合法构造。用链表维护交界处，时间复杂度 $O(n^2)$。

二分图匹配

算法大家都很熟悉了，不再赘述。

都知道最大匹配等于最小点覆盖，最小点覆盖怎么据此构造？

从左侧非匹配点出发，向右走非匹配边，向左走匹配边，最后取左侧未访问到的点 + 右侧访问到的点。用最小割模型很好理解。

二分图匹配的必经/可行的点/边，同样可以通过最小割帮助判断。

二分图匹配

算法大家都很熟悉了，不再赘述。

都知道最大匹配等于最小点覆盖，最小点覆盖怎么据此构造？

从左侧非匹配点出发，向右走非匹配边，向左走匹配边，最后取左侧未访问到的点 + 右侧访问到的点。用最小割模型很好理解。

二分图匹配的必经/可行的点/边，同样可以通过最小割帮助判断。

二分图匹配

算法大家都很熟悉了，不再赘述。

都知道最大匹配等于最小点覆盖，最小点覆盖怎么据此构造？

从左侧非匹配点出发，向右走非匹配边，向左走匹配边，最后取左侧未访问到的点 + 右侧访问到的点。用最小割模型很好理解。

二分图匹配的必经/可行的点/边，同样可以通过最小割帮助判断。

Hall 定理

描述：二分图有完美匹配，当且仅当对任意左部点的子集 S，右部点中与 S 相邻的点数不小于 $|S|$。

证明略。

扩展：

- 二分图最大匹配为 $N - \max_S\{|S| - |N(S)|\}$。
- 二分图匹配在矩阵中有代数意义，具体可见后面的 LGV 引理。

一个常见的模型是点向区间连边，这时枚举子集可以转化为枚举区间。

有时也可以反过来用，将诸如上述的限制转化到二分图上。

ARC076F Exhausted?

题目描述

有 m 张椅子与 n 个人，第 i 张椅子在坐标 i，第 i 个人可坐的椅子坐标范围为 $[1, l_i] \cup [r_i, m]$。
求最少加多少把椅子使得每个人都有椅子坐。
$n, m \le 2 \times 10^5$。

Sol

根据 Hall 定理的推论，我们需要求的就是 $\max_S\{|S|-|N(S)|\}$。显然 $N(S)$ 一定为前后缀并，本质不同的 $N(S)$ 只有 m^2 种。

这启示我们转而枚举 $N(S)$ 的具体情况。具体的，设 $N(S)=[1,l]\cup[r,m]$，我们希望 $|S|$ 越大越好，肯定是把 $l_i\le l,r_i\ge r$ 的人全部选上。由于我们只要求 $\max$，因此实际的并集 $\subseteq N(S)$ 也没有关系。

使用扫描线 $+$ 线段树即可维护。

一般图匹配与 Tutte 矩阵

似乎超纲了，不过形式很简单，就听一听吧。

对图 G，定义 Tutte 矩阵 A 满足：

$$A_{ij} = \begin{cases} \lambda_{i,j}, & (i,j) \in G \\ -\lambda_{i,j}, & (i,j) \in G \\ 0 & \text{otherwise} \end{cases}$$

注意每个 $\lambda_{i,j}$ 都是独立的变元。则 G 存在完美匹配 $\iff |A| \neq 0$。

证明考虑有偶环覆盖 $\iff$ 有完美匹配。

实际应用中一般将 $\lambda_{i,j}$ 随机赋权，在 $\bmod P$ 意义下计算行列式。

一般情况，可以利用线性代数知识证明最大匹配 $= \frac{\mathrm{rank}(A)}{2}$。

构造方案：直接尝试每条边是否在最大匹配中。直接做是 $O(n^4)$ 的，容易使用线性代数知识优化至 $O(n^3)$。

一般图匹配与 Tutte 矩阵

似乎超纲了，不过形式很简单，就听一听吧。

对图 G，定义 Tutte 矩阵 A 满足：

$$A_{ij} = \begin{cases} \lambda_{i,j}, & (i,j) \in G \\ -\lambda_{i,j}, & (i,j) \in G \\ 0 & \text{otherwise} \end{cases}$$

注意每个 $\lambda_{i,j}$ 都是独立的变元。则 G 存在完美匹配 $\iff |A| \neq 0$。

证明考虑有偶环覆盖 $\iff$ 有完美匹配。

实际应用中一般将 $\lambda_{i,j}$ 随机赋权，在 $\bmod P$ 意义下计算行列式。

一般情况，可以利用线性代数知识证明最大匹配 $= \frac{\operatorname{rank}(A)}{2}$。

构造方案：直接尝试每条边是否在最大匹配中。直接做是 $O(n^4)$ 的，容易使用线性代数知识优化至 $O(n^3)$。

一般图匹配

这里还有一个更加简洁、正确性较高的随机算法。

注意到一般图不能直接像二分那样增广是因为有奇环，所以反转匹配状态会导致冲突。考虑强制不找环。只找长为奇数的链进行增广。本质上相当于对奇环断环为链。

但这样从每个点出发跑一遍尝试增广的错误率很高，那就随机顺序多跑几次增广即可。

学有余力可以直接去学较为复杂的带花树算法。

一般图匹配

这里还有一个更加简洁、正确性较高的随机算法。

注意到一般图不能直接像二分那样增广是因为有奇环，所以反转匹配状态会导致冲突。考虑强制不找环。只找长为奇数的链进行增广。本质上相当于对奇环断环为链。

但这样从每个点出发跑一遍尝试增广的错误率很高，那就随机顺序多跑几次增广即可。

学有余力可以直接去学较为复杂的带花树算法。

二分图博弈

考虑这样一个问题：

A,B 在二分图上轮流移动棋子，初始在 s，每次可把棋子沿边移动到一个尚未访问的点，不能移动者输。

结论

先手必胜当且仅当 s 一定在最大匹配上。

证明考虑增广路即可。可以推广至一般图。

一个常见模型是结合网格图的黑白染色。

二分图博弈

考虑这样一个问题：

A,B 在二分图上轮流移动棋子，初始在 s，每次可把棋子沿边移动到一个尚未访问的点，不能移动者输。

结论

先手必胜当且仅当 s 一定在最大匹配上。

证明考虑增广路即可。可以推广至一般图。

一个常见模型是结合网格图的黑白染色。

NOI 2011 兔兔与蛋蛋游戏

题目描述

$n \times m$ 的棋盘上有黑白两色棋子，且仅有一个空格。
A 可以将空格相邻的黑棋移入空格中，B 可以将空格相邻的白棋移入空格中。
不能移动者输。给出双方操作序列，判断 A 有几次操作失误，即操作前 A 有必胜策略、操作后 B 有必胜策略。
$n, m \leq 40$。

然而这道题并不是黑白染色...

NOI 2011 兔兔与蛋蛋游戏

题目描述

$n \times m$ 的棋盘上有黑白两色棋子，且仅有一个空格。
A 可以将空格相邻的黑棋移入空格中，B 可以将空格相邻的白棋移入空格中。不能移动者输。给出双方操作序列，判断 A 有几次操作失误，即操作前 A 有必胜策略、操作后 B 有必胜策略。
$n, m \leq 40$。

然而这道题并不是黑白染色...

Sol

把棋子移动视为空格移动，则相当于空格选择一条黑白交替的路径在移动。

这样，二分图的模型就出来了。把黑棋、白棋的位置视为二分图左右两部，相邻的黑棋白棋连边，这就成为了二分图博弈模型。

至于判断失误，预处理每轮游戏先手是否有必胜策略即可，利用匈牙利算法即可判断每次是否在最大匹配必经点上。

Dilworth 定理

在一个**偏序集**的**传递闭包**中，最小链覆盖 $=$ 最长反链。证明考虑归纳即可。
扩展：最小反链覆盖 $=$ 最长链。

LGV 引理

用来处理 DAG 上的不相交路径计数。

具体形式

设 S 为起点集合，T 为终点集合，$|S| = |T| = n$。
设 P_i 为一条路径，$f(P_i)$ 为路径沿途各边权值乘积，$f(x, y)$ 为从 x 到 y 所有路径沿途各边权值乘积之和。
对一个路径组 P，P_i 从 S_i 出发到 $T_{\pi(i)}$，设 $f(P) = \prod_{i=1}^{n} f(P_i)$。
则令

$$M = \begin{pmatrix} f(S_1, T_1) & f(S_1, T_2) & \cdots & f(S_1, T_n) \\ f(S_2, T_1) & f(S_2, T_2) & \cdots & f(S_2, T_n) \\ & & \vdots & \\ f(S_n, T_1) & f(S_n, T_2) & \cdots & f(S_n, T_n) \end{pmatrix}$$

有 $|M| = \sum_{P\text{为路径组}} (-1)^{\sigma(\pi)} f(P)$。

LGV 引理

证明分开考虑有交的路径组和无交的路径组，发现有交的路径组可两两配对并抵消。

一些扩展应用：

- 如果只关心路径组的存在性，为了排除 $(-1)^k$ 项的影响，可以考虑对边随机权值，在模意义下运算，最后判断行列式是否非空。
- 可以推广至求 S 到 T 的最大匹配：由线性代数知识，这即为矩阵的秩。在此，二分图匹配可以视为一个特例——这体现了二分图匹配的代数意义。

Unknown Source

题目描述

给定分层图，共 n 层、每层有 k 个顶点。q 次询问，每次询问第 i 层点集 S 到第 j 层点集 T 的最小割点集，即最少删除多少个点使得从 S 任意点出发不能到 T 中任意点。

$n, q \leq 5000, k \leq 30$。Bonus：带修。

这展示了 LGV 引理与分层图最大流/最小割的关系。

Unknown Source

题目描述

给定分层图，共 n 层、每层有 k 个顶点。q 次询问，每次询问第 i 层点集 S 到第 j 层点集 T 的最小割点集，即最少删除多少个点使得从 S 任意点出发不能到 T 中任意点。
$n, q \leq 5000, k \leq 30$。Bonus：带修。

这展示了 LGV 引理与分层图最大流/最小割的关系。

Sol

最小割点集就是从 S 到 T 的最多不相交路径数。

对于分层图来说，路径条数就是邻接矩阵的乘积。

求出第 i 层到第 j 层邻接矩阵的乘积，高斯消元求秩即为答案。

至于带修，用线段树维护矩阵乘积即可。

感谢倾听。