

图论

Wearry

Stay determined!

① 连通性与最短路

② 生成树与生成图

③ 度数与回路

④ 匹配与网络流

故乡的梦

给出一个 n 个点 m 条边的无向图，边权非负， q 次询问删除一条边之后 s, t 之间的最短路长度。

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

故乡的梦

首先分别求出从 s, t 出发的单源最短路，并建出最短路图，包含所有满足 $diss_u + w(u, v) = dist_v$ 的边 (u, v) ，忽略所有孤立点，可以发现这个新图缩完边双之后将形成一条链。

考虑询问的边，发现如果这样的边不是新图的桥，一定不会影响最短路的长度，否则可以离线处理删除任意两个相邻的边双之间桥的答案。

Travel

给定一个 n 个点 m 条边的无向联通图，每个点有点权，接下来会进行 q 次如下两种操作：

- ① 修改某个点的点权
- ② 询问从 u 出发经过简单路径到达 v 可能经过的点中点权的最小值

$$n, m, q \leq 10^5$$

Travel

可以把将原图的点双联通分量缩点得到一棵树，如果不存在修改操作，就相当于求树上两点之间的路径上的最小点权。

修改的时候只需要对于每个点双维护 `multiset` 即可，需要注意的是割点可能在多个点双内部。可以在修改的时候，只考虑割点的父亲节点所在的点双，询问的时候特殊考虑割点的贡献。

[POI 2004] ZAW

给定一个 n 个点 m 条边的无向图，边权 $w(u, v)$ 和 $w(v, u)$ 可能不同。
求一个包含 1 号点的边权和尽可能小的有向环。

$$n \leq 3 \times 10^4, m \leq 10^5, 1 \leq w \leq 10^4$$

[POI 2004] ZAW

考虑枚举从一号点出发的边 $(1, x)$ 那么环上剩下的部分就是 $dis(x, y) + w(y, 1)$, 然而这样需要对所有可能的 x, y 都计算一遍答案.

实际上每次可以处理多组这样的 x, y : 将 X 集合中的点的初始距离设为 $w(1, x)$, 并将从 1 号点连向 X 集合中点的边删除, 然后跑一次多源最短路, 求出 $\min\{dis(y) + w(y, 1)\}$ 即可.

利用二进制的性质来划分集合, 每次将二进制下编号第 i 位为 0 和 1 的点划分为 X, Y 两个集合, 这样分组的次数仅为 $O(\log N)$.

[UR 2] 跳蚤公路

给出一个 n 个点 m 条边的有向图，其中有三类边，边权分别为：

- w_i
- $w_i + x$
- $w_i - x$

对于每个点，求满足从 1 号点出发到此点的路径上均不存在负权环的 x 的取值范围。

$$n \leq 100, m \leq 10^4$$

[UR 2] 跳蚤公路

在使用 Bellman-Ford 算法判断负权环的过程中，如果记 $f_{i,u}$ 表示从起点出发经过不超过 i 步到达 u 的最短距离，那么图上有负环的就等价于存在某个点 u ，使得 $f_{n-1,u} > f_{n,u}$ ，并且这样的 u 一定在负环上。

[UR 2] 跳蚤公路

在使用 Bellman-Ford 算法判断负权环的过程中，如果记 $f_{i,u}$ 表示从起点出发经过不超过 i 步到达 u 的最短距离，那么图上有负环的就等价于存在某个点 u ，使得 $f_{n-1,u} > f_{n,u}$ ，并且这样的 u 一定在负环上。

在这个问题中，可以记 $f_{i,j,u}$ 表示从起点出发经过不超过 i 步到达点 u ，且 x 前的系数为 j 时路径长度的常数部分的最小值，存在负环的条件与上述类似：

$$\min_i \{f_{n-1,i,u} + ix\} > \min_j \{f_{n,j,u} + jx\}$$

这样对于某个点的合法的 x 的取值范围则相当于所有能到达它的点均不在负环上的解集的交。

Wander

在一个无向联通图上跑 SPFA 算法，初始时每个点的距离为 10^{18} ，给定跑完后得到的队列，要求构造一个原图满足要求。

$$n \leq 10^5, |q| \leq 10^6$$

Wander

显然每个点最后一次出现在队列中时最短路长度会确定下来，同时可以假设使用 *Dijkstra* 算法得到了队列，对于重复出现在队列中的点，应当认为除了最后一次之外都无法对其他点的距离进行更新。

这样显然不影响最短路长度的正确性，尝试按照如下方法构造：

- ① 假设某个点除最后一次外第 k 次出现在队列中时它的距离为 $10^{18} - k$
- ② 记某个点 x 在队列中是第 r_x 个变成最短长度的点，当某个点最后一次入队后它的最短路长度变成 $r_x \times n^2$

① 连通性与最短路

② 生成树与生成图

③ 度数与回路

④ 匹配与网络流

[BZOJ 2521] 最小生成树

对于一个无向图，可以进行若干次操作，每次操作能够使得图上某一条边的权值加上 1 .

求至少要进行多少次操作能够使得边 (u, v) 一定在图的最小生成树上.

$$n \leq 500, m \leq 800$$

[BZOJ 2521] 最小生成树

根据最小生成树的性质，某条边 (u, v) 一定在最小生成树上等价于从 u 出发到 v 的所有路径的最大边的边权大于 $w(u, v)$.

将每条边都重新设置边权为：

$$w' = \max\{w(u, v) - w + 1, 0\}$$

则原问题转化为求 u 到 v 的最小割.

Flare

定义简单无向图 G 的转置 G^T :

- G^T 上的每个点对应图 G 中的一条边.
- G^T 上的两点间有边当且仅当在 G 中对应的两条边有公共端点.

现在已知某个 G^T , 求任意一个可能的 G 或说明不存在合法的 G .

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

Flare

假设新图中的某个点 u 在原图中对应的边是 (a, b) ，那么它的邻域 $N(u)$ 中的点对应的边一定与 a, b 中的一个点相连。

所有原图中与点 a 相连的边在新图上会形成一个团。

而且这样子的二分图上的每一个点一定最多只与另一侧的一个点不相连，即原图每个点连出的杂边只有最多一条。

Flare

这个图的补图是一个二分图，可以先用染色法判断二分图，如果判断失败则无解，否则假设已经确定的二分图分为 X, Y 两个部分。

讨论如下的几种情况：

- $|X| = 1, |Y| = 0$ 即存在孤立点，则应该考虑换一个起点重新染色，如果所有点都是孤立点并且数量大于 2，则一定均属于 X
- $|X| = 1, |Y| = 1$ 未确定的点不能超过两个
- $|X| = 1, |Y| > 1$ 所有未确定的点都与 Y 在二分图的同侧
- $|X| > 1, |Y| > 1$ 一定不能有未确定的点

Flare

这样未确定集合的点一定不超过两个，于是可以枚举它们所属的集合，然后进行一次 BFS 确定图中其他的所有点的编号即可。

一开始每个点的标号都是 $(?, ?)$ 染色过的点的编号为 $(a, ?)$ 或者 $(b, ?)$ ，每次选择一个确定了一半的点 $(x, ?)$ 并更新相邻点的信息：

- 如果存在相邻的 $(y, ?)$ ，则这两个点都应该是 (x, y) ，于是不合法
- 否则新建编号 z ，令这个点为 (x, z) 并更新所有其他 $(?, ?)$ 点为 $(z, ?)$

① 连通性与最短路

② 生成树与生成图

③ 度数与回路

④ 匹配与网络流

Palingenesis

统计 n 个点 m 条边的无向图的四元环数量.

$$n \leq 5 \times 10^4, m \leq 10^5$$

Palingenesis

性质：对于图上的每一个点，与其相邻且度数不小于它的点数量是 $O(\sqrt{m})$ 的。

将图中的所有点按照度数排序，并且将所有的边定向成从度数较小的点连向度数较大的点，这样四元环的形式只有两种，统计的时候可以在度数最大的点处断开成两个部分，每个部分都包含两条边，可以分别处理。

这样可以做到 $O(m\sqrt{m})$ 的复杂度。

Bus stop

有一个 n 个点 $2n$ 条边，每个点的入度出度均为 2 的有向图，求图中最短的欧拉回路，回路的代价与经过每个点时的边的顺序有关。

$$n \leq 10^5$$

Bus stop

不妨先考虑一个贪心的思路，首先在每个点上选择代价最小的方案，得到初始的一个回路，不难发现由于每个点的入度都等于出度，因此这个初始的回路由若干个独立的环构成。

接下来考虑更改某个点的方案，如果变更方案之前这个点在两个不同的环上，那么方案变更后两个环就会合并。

题目需要将所有的环合并为一个，用并查集维护这些环，则可以贪心地合并这些环，算法过程类似最小生成树的 Kruskal 算法。

- ① 连通性与最短路
- ② 生成树与生成图
- ③ 度数与回路
- ④ 匹配与网络流

常见模型

最小点覆盖

选择最少的点，使得每条边至少有一个端点在被选择点的集合中

最小边覆盖

选择最小的边，使得每个点至少有一条邻边在被选择边的集合中

最大独立集

选择尽可能多的点，使得它们两两不相邻

最大权闭合子图

在一个点带正权或者负权的有向图上选择一些点使得总权值尽可能大，其中边 (u, v) 表示选择点 u 就必须选择点 v

常见模型

最小路径覆盖

用尽可能少的路径覆盖每个点恰好一次，路径是一系列相连的点构成的集合

最小链覆盖

用尽可能少的链覆盖每个点恰好一次，链是一个点的集合满足其中任意两个点 u, v 满足 u 能到 v 或者 v 能到 u

最长反链

求一个尽可能大的点的集合满足集合内的点两两无法到达

常用结论

设二分图点数为 n , 最大匹配数为 m

最小点覆盖 = m

由于不存在增广路, 找到最边缘的匹配边并依次选择匹配边同侧的点即可

最小边覆盖 = $n - m$

匹配边以外的其他边一定连接一个匹配点与一个未配点

最大独立集 = $n - m$

独立集的补集是点覆盖集

最大权闭合子图

将源点连向正权点, 负权点连向汇点, 边权均为点权的绝对值, 最后用所有正权点的权值和减去最小割即得到答案

常用结论

最小路径覆盖

将每个点 u 拆成两个点 u_0, u_1 , 对于原图中的每条边 (u, v) 连一条 (u_0, v_1) 的边, 最后用原图中点的个数减去最大匹配数即可

最小链覆盖

先在原图上做一个传递闭包, 然后做一次最小路径覆盖

最长反链 = 最小链覆盖

Dilworth 定理

二分图匹配的关键点和关键边

- 关键点：先求一个最大匹配方案，然后从未盖点出发做 BFS，遍历到的被覆盖的点都是非关键点
- 关键边：关键边的两个端点均为关键点且不在图中的偶环上

Chess

给定一个 4×4 的棋盘，棋盘上有一些障碍、空地和棋子，两个玩家 eyu 轮流进行操作，每次操作的时候可以将一个棋子移动到一个相邻的空地上，或者移除一个棋子。不能操作的人输。

为了防止出现平局，不允许某次操作后出现之前已经出现过的局面。

现在给出了若干初始局面，要你判断每个局面是否先手必胜。

Chess

首先可以将所有的局面按照棋子的个数分组，对于棋子个数相同的状态，其形成一个二分图.

如果某个状态向下一层的状态转移能够先手必胜，那么这个状态就是先手必胜的；对于其他状态，不难发现它们构成的二分图关键点一定是必胜点，非关键点是必败点.

[BZOJ 1937] Mst 最小生成树

给出一个 n 个点 m 条边的带权无向图和一个它的生成树，每次操作可以将一条边的权值加一或者减一，问要使得这个生成树变为原图的最小生成树（不要求唯一），至少要进行多少次操作。

$$n \leq 50, m \leq 800, w_i \leq 1000$$

[BZOJ 1937] Mst 最小生成树

显然操作的时候一定会减小生成树上边的权值，增加非树边的权值，对于每条非树边的要求是满足权值大于等于其覆盖的所有树边权值。

假设现在有一条树边与一条非树边 x, y 则要求：

$$\begin{aligned} w_x - d_x &\leq w_y + d_y \\ d_x + d_y &\geq w_y - w_x \end{aligned}$$

不难发现这个要求与 KM 算法中的可行顶标是类似的，同时我们知道：使用 KM 算法求得的结果中可行顶标之和一定是最小的

[BZOJ 2229] [ZJOI2011] 最小割

给出一个 n 个点 m 条边的无向图， q 次询问，每次询问给定一个 x ，要求回答图中有多少对点的最小割值小于等于 x .

无向图上两点 s, t 的最小割定义为一个边权和最小的边集 E ，满足删除 E 中的点后 s, t 不联通，根据删掉所有边后 s 和 t 所在的位置将原图分为 s 部和 t 部两个部分.

$$n \leq 150, m \leq 3000, q \leq 30$$

[BZOJ 2229] [ZJOI2011] 最小割

考虑使用最小割树来解决这个问题，最小割树的构造方法是，每次任意选择两个点 s, t ，将 (s, t) 的边权设为这两点之间的最小割值，然后将 s 部和 t 部分别向边的两侧递归构造即可。

这样任意两点之间 u, v 的最小割就等于它们在最小割树的路径上边权最小的边。

[BZOJ 2229] [ZJOI2011] 最小割

接下来考虑证明这样做的正确性，记 u, v 两点的最小割为 $\lambda(u, v)$:

- $\lambda(a, b) \geq \min\{\lambda(a, c), \lambda(b, c)\}$
- $\lambda(u, v) \geq \min\{\lambda(u, x_1), \lambda(x_1, x_2) \cdots \lambda(x_k, v)\}$

修路

给出一个二分图，要求选出其中的一些边使得：

- ① 每个点的度数至少是 1，至多是 k
- ② 选出来的边不存在环且联通块数量尽可能多
- ③ 选出的边权和最大

$$n \leq 200, m \leq 2000$$

修路

不难发现条件二本质上是要求边的数量尽可能少，于是可以给每条边的权值都减掉 ∞ .

然后源点向左部的每个点，右部的每个点向汇点连一条下界为 1，上界为 k 的边，然后求上下界最大费用流即可.