《Minimal Cut》解题报告

重庆市南开（融侨）中学 何宇翔

目 录

1 题目链接

https://qoj.ac/contest/532/problem/892

2 题目描述

给一张 n 个点 $m+n$ 条边的图，边有边权 w。其中第 $i(1 \le i \le n)$ 条边连接点 i 与 $(i \bmod n)+1$，边权为 10^9。其他边与边权输入。问两两点之间最小割之和是多少。

$n, m \le 2 \times 10^4, 1 \le w \le 10^4$。

3 解题过程

3.1 最小割树

求两两点间最小割是经典问题，可以用最小割树解决。求最小割树有算法 Gomory-Hu。算法流程如下：

维护一棵 n 个点的最小割树，最初没有任何边。执行如下函数，传入一个点集 S，最初调用全集。

1. 若 $|S| = 1$，则结束递归。
2. 任选 S 中两个不同的点，设为 x, y，在原图上求出它们之间的最小割 $cut(x, y)$。
3. 设在最小割中 x, y 所在点集分别为 U, V，则递归调用 $S \cap U, S \cap V$。
4. 设 p 为调用 $S \cap U$ 时连边的两点中，所在点集包含 y 的点，q 为调用 $S \cap V$ 时连边的两点中，所在点集包含 x 的点。如果调用的集合大小为 1 则直接等于那个唯一的元素。在最小割树上连接 p, q 边权为 $cut(x, y)$。

最终会计算 $n-1$ 次最小割，建的边构成一棵树。最终任意两点间的最小割即为路径上边权的最小值。

这里的连边方式不同于一般地直接连接 x, y，因为这样建出的最小割树有更强的性质，接下来证明两种方式的正确性。

证明. 先给出结论，按照上述方法连成的最小割树，任意两点 x,y 路径上所有边对应的割都能割开 x,y。

设 $Cut(A)$ 表示所有恰有一个端点在 A 中的边的边权和。根据定义我们可以得到：

$$Cut(A)+Cut(B)\geq Cut(A\cap B)+Cut(A\cup B)$$

设某次所选两点 a,b 的最小割 $cut(a,b)$ 中 a,b 所在点集分别为 A,B。$u,v\in A$ 的最小割 $cut(u,v)$ 中 u,v 所在点集分别为 U,V。b 一定属于 U,V 之一，不妨设 $b\in V$。则 $Cut(A\cup U)$ 是 a,b 的一个割，而 $Cut(A\cap U)$ 是 u,v 的一个割。所以：

$$Cut(A)\leq Cut(A\cup U)$$
$$Cut(U)\leq Cut(A\cap U)$$

结合之前的结论可以得到 $Cut(U)=Cut(A\cap U)$。

再观察函数中的建边方式，如果调用 A 集合时所选的点就为 u,v，所连的边为 p,q，其中 $p\in U$ 而 $q,b\in V$，我们让 q 成为 p 的父亲。该次函数我们会将 q 与 B 集合中某点连 $cut(a,b)$ 的边。

根据描述，除了最后连边的两点，其余点恰有一个父节点，不看最后一条边则形成两棵有根树，设 T_x 为有根树上的子树点集，则 $Cut(A\cap U)=T_p$。于是：

$$cut(u,v)=Cut(U)=Cut(A\cap U)=T_p$$

得到结论：最小割树上的一条边将树划分出两个点集就是对应割的一种划分。自然有 x,y 路径上所有边对应的割能割开 x,y。

而任意两点间最小割不可能比路径上的边权都小，否则这个割就会成为一条边了。于是原图任意两点最小割就是这棵树的最小割，甚至划分出的点集也是相同的，这棵树可以直接替代原图。我们之后称其为理想最小割树。

我们再证明另一种连接方式的正确性，即在函数第 4 步时，改为连接 x,y 边权为 $cut(x,y)$。称这样连出来的图为一般最小割树。借助理想最小割树，正确性是比较明了的。

将求最小割操作替换为在理想树上找到路径最小值。则过程为：每次取两个点连边，边权为理想树上路径最小值，再分别完成理想树上边两侧的子问题。则任意两点在理想树上的路径最小值肯定会出现在一般树的路径上，而不在理想树路径上且比路径最小值更小的割肯定不会在一般树的路径上。所以，一般最小割树求任意两点最小割是正确的，但是不能快速给出划分方式。

□

3.2 算法 1

借助最小割树，我们只需要在静态图上在线地求 $n-1$ 次两点间最小割即可。直接跑网络流没有优化空间，可以分析图的性质。

称前 n 条边权为 10^9 的边是环边，其余是非环边。

发现 $m \times \max(w) < 10^9$，意味着要断掉尽量少的环边，也就是两条。源汇点会将环分成两侧，在每侧都枚举一条环边断掉，此时环会断成两部分，非环边若连接了两部分则也要断掉，这样就是一个割了。

使用数据结构可以更快地完成一次最小割，把两侧的边看作坐标的两个维度，两侧割掉的边可以定位平面的一个点，一条非环边会使得 $O(1)$ 个矩形增加代价，最小割即为平面内代价最小点的代价。扫面线并维护线段树可以做到 $O((n+m)\log n)$ 求一次。

3.3 算法 2

每次求最小割时都跑一遍上述方法显然不可取，要避免每次重新建数据结构来维护平面。分析发现，无论源汇点是什么，若一条非环边在两条环边异侧则需要割掉，否则不用。于是我们建一个全局的平面，每条边会让 $O(1)$ 个矩形增加代价。询问时根据源汇点，查询的两维都在一个范围内，拆成 $O(1)$ 个矩形查询最小值操作。

现在问题为 $O(m)$ 次矩形加操作后，在线地进行 $O(n)$ 次矩形查最小值。可以通过猫树分治来解决。按照某一维分治，每次从分治中心向两侧扫描，并维护区间历史最小值即可。时间复杂度 $O(m\log^2 m + n\log n)$。不过由于询问在线，需要可持久，空间复杂度也是 $O(m\log^2 m)$ 还带大常数。

再次观察查询的形式，如果源汇点是 $l, r(l<r)$，需要查询矩形 $[1,l)\times[l,r), [l,r)\times[r,n]$。一定有一维是前缀或者后缀，所以只需要维护从前往后以及从后往前的区间历史最值即可，同样要可持久。维护的时空复杂度都降为 $O(m\log m)$。可以通过此题。