浅谈树上邻域与链邻域的并交问题

杭州第二中学 汪苏轶

摘 要

本文介绍了树上邻域理论，树上邻域求并与求交方法，以及树上链邻域求交的理论与算法，并给出了这些方法在相关问题中的一些应用。

引言

树结构相关问题是算法竞赛中的重要研究对象，其中树上邻域类问题亦十分常见。

由树上邻域的概念出发，可以自然地类比到平面几何中的圆域问题。经过研究发现，树上邻域在进行并与交等运算时，呈现出与平面圆域相似的结构性与美感；进一步推广至树上链邻域后，这些良好的结构性质在一定程度上仍得以保留。

本文将围绕树上邻域的并、交运算以及树上链邻域的交运算展开讨论，并结合具体问题，对相关性质进行简单的的分析与探讨。

1 记号与约定

设 $T = (V, E)$ 是一张无向图，其中 V 为非空顶点集合，E 为边集合。称 T 是一棵树，当且仅当 T 联通且无环。

树 T 上任意两点之间的简单路径与距离可按如下方式定义。

定义 1.1 (简单路径)**.** 对于任意 $s, t \in V$，定义其简单路径为

$$\mathrm{Path}^T(s,t) = (s = v_0, v_1, \ldots, v_k = t),$$

其中 $v_i \in V$，且满足：

1. 对所有 $0 \le i < k$，$(v_i, v_{i+1}) \in E$；
2. $v_0, v_1, \ldots, v_k$ 两两不同。

在树中，任意两点间的简单路径唯一。也称 s, t 之间的唯一简单路径为连接 s, t 的链。

定义 1.2 (距离)**.** 若 $\mathrm{Path}^r(s,t)=(v_0,\ldots,v_k)$，则称

$$d_T^r(s,t)=k$$

为 s 与 t 的树上距离。

为了描述树上的边中点等结构，尝试扩展原有点集。

定义 1.3 (广义点)**.** 定义广义点集合

$$\mathcal{V}=V\cup\{\mathcal{M}(u,v)\mid(u,v)\in E\},$$

其中 $\mathcal{M}(u,v)$ 表示边 (u,v) 的中点。

定义 1.4 (广义距离)**.** 先定义基本距离 $d_e:\mathcal{V}\times V\to\mathbb{R}\cup\{+\infty\}$：

$$d_e(u,x)=\begin{cases}0, & u=x,\\ \\ +\infty, & \text{otherwise},\end{cases}$$

$$d_e(\mathcal{M}(u,v),x)=\begin{cases}\frac{1}{2}, & x=u\text{ 或}x=v,\\ \\ +\infty, & \text{otherwise}.\end{cases}$$

广义距离 $dis:\mathcal{V}\times\mathcal{V}\to\mathbb{R}$ 定义为

$$dis(p,q)=\min_{x,y\in V}(d_T^r(x,y)+d_e(p,x)+d_e(q,y)).$$

定义 1.5 (广义路径)**.** 定义广义路径为

$$\mathrm{Path}(p,q)=\{x\in\mathcal{V}\mid dis(p,x)+dis(x,q)=dis(p,q)\}.$$

表示 p 与 q 在树上路径中经过的所有广义点。

定义 1.6 (行走)**.** 定义行走函数 $\curvearrowright:\mathcal{V}\times\mathcal{V}\times\mathbb{R}\to\mathcal{V}$，其中 $\curvearrowright(s,t,\theta)$ 表示从 s 朝向 t 行走距离 θ 后到达的唯一点（$0\le\theta\le d_T(s,t)$）。形式化为：

$$\curvearrowright(s,t,\theta)=p\quad\text{where }p\in\mathrm{Path}(s,t),\ d_T(s,p)=\theta.$$

上述关于广义点的相关定义仅为了辅助接下来树上邻域理论的描述，其基本保留了传统树上路径、距离等的性质。

定义 1.7 (树上邻域)**.** 定义广义点 u 在树上的 r 邻域为所有与 u 之距离 $\le r$ 的点所构成的集合，记作 $N(u,r)$。形式化的，有：

$$N(u,r)=\{x\in\mathcal{V}\mid dis(u,x)\le r\}$$

称 u 为邻域 $N(u,r)$ 的中心，r 为该邻域的半径。

称一个邻域 $N(u,r)$ 是真实的，当且仅当满足以下两个条件之一：

$$\begin{cases} u \in V, r \in \mathbb{Z} \\ u \notin V, r = p + \frac{1}{2}, p \in \mathbb{Z} \end{cases}$$

可理解为该邻域的所有边界点都是树上真实的顶点。

本文中所讨论的所有邻域均为真实邻域。接下来如果未经特殊说明，所有的邻域均指真实邻域。

定义 1.8 (点到链的距离)**.** 对于树上的顶点 u 和一条连接顶点 s,t 的链 $\text{Path}(s,t)$，定义 u 到 $\text{Path}(s,t)$ 的距离：

$$dis(u, \text{Path}(s,t)) = \min_{x \in Path(s,t)} dis(u,x)$$

同时，若上述式子中取到最小值的 $x = x_0$，则称 u 属于 $\text{Path}(s,t)$ 的 x_0，或者 u 在链 $\text{Path}(s,t)$ 上属于 x_0。

定义 1.9 (链到链的距离)**.** 对于树上任意两条链 $\text{Path}(u_1,v_1)$ 和 $\text{Path}(u_2,v_2)$，定义这两条链之间的距离：

$$dis(\text{Path}(u_1,v_1), \text{Path}(u_2,v_2)) = \min_{x \in \text{Path}(u_1,v_1)} \min_{y \in \text{Path}(u_2,v_2)} dis(x,y)$$

定义 1.10 (树上链邻域)**.** 定义一条连接顶点 u,v 的链的 r 邻域为所有与链 $\text{Path}(u,v)$ 距离 $\le r$ 的点所构成的集合，记作 $M(u,v,r)$。形式化的，有：

$$M(u,v,r) = \{x \in V \mid dis(x, \text{Path}(u,v)) \le r\}$$

类似的，称 $\text{Path}(u,v)$ 为链邻域 $M(u,v,r)$ 的中心链，r 为该链邻域的半径。

本文中讨论的所有有关树上链邻域的问题均不包含广义点。

2 树上邻域理论

2.1 直径与树上邻域相关性质

本小节将首先研究一些有关树上邻域的基本性质。

由树上邻域的定义 $N(u,r)$ 不难联想到平面上的圆 $circle(o,r)$，故树上邻域在一些场合也被称之为“树上圆”。对于平面上的一个圆，连接圆内任意两点的最长直线即为直径，而所有的直径都会经过圆心一点。由此容易引导将该结论类比到树上。

对于树上的一个由顶点构成的点集 S，定义点集 S 的直径长度 $d(S)$ 为 $\max_{x,y \in S} dis(x,y)$，而任意一条满足 $x,y \in S$ 且 $dis(x,y) = d(S)$ 的路径 $\text{Path}(x,y)$ 都被称为点集 S 的直径。有如下定理：

定理 2.1. 对于树上任意一个点集 S，S 的所有直径中点重合（中点可能为广义点）。

证明. 反证。若存在 S 的两条不同直径 $\mathrm{Path}(x_1, y_1)$ 和 $\mathrm{Path}(x_2, y_2)$ 使得两条直径的中点 $c_1 \neq c_2$，则 $\mathrm{Path}(c_1, x_1)$ 和 $\mathrm{Path}(c_1, y_1)$ 中比有一条路径与 $\mathrm{Path}(c_1, c_2)$ 只在 c_1 相交，不妨设为 $\mathrm{Path}(c_1, x_1)$；同理有 $\mathrm{Path}(c_2, x_2)$。那么 $dis(x_1, x_2) = dis(x_1, c_1) + dis(x_2, c_2) + dis(c_1, c_2) = d(S) + dis(c_1, c_2)$，与直径的定义中的最大性不符，矛盾。 □

故对于一个点集 S，可以定义点集的中点 $m(S)$ 为一个广义点。找到中点和直径后，就可以类似平面上的圆，定义点集 S 对应的树上邻域 $\mathcal{K}(S) = N(m(S), \frac{d(S)}{2})$。容易发现 $\mathcal{K}(S)$ 必然是一个真实邻域。

接下来考虑一些从平面中的圆覆盖所引导出的性质：

定理 2.2. 对于树上任意一个点集 S 与真实邻域 $N(u, r)$，若 $S \subset N(u, r)$，则 $\mathcal{K}(S) \subset N(u, r)$。

证明. 首先证明 $dis(u, m(S)) \leq r - \frac{d(S)}{2}$。若不然，则取任意一条 S 的直径 $\mathrm{Path}(x, y)$，必有 x, y 中的一者在 $m(S)$ 的与 u 不同的一棵子树中。不妨设 x 是，那么由 $x \in S \subset N(u, r)$ 可知 $dis(x, u) = dis(x, m(S)) + dis(m(S), u) = \frac{d(S)}{2} + dis(m(S), u) \leq r$，因此 $dis(m(S), u) \leq r - \frac{d(S)}{2}$。

从而对于任意一点 $t \in CC(S) = N(m(S), \frac{d(S)}{2})$，都有 $dis(t, u) \leq dis(t, m(S)) + d(m(S), u) \leq \frac{d(S)}{2} + (r - \frac{d(S)}{2}) = r$，因此 $t \in N(u, r)$。进而可得 $\mathcal{K}(S) \subset N(u, r)$。 □

2.2 树上邻域求并

在讨论平面上的圆求并问题时，通常会考虑求解最小圆覆盖。即用一个最小的圆覆盖所有给定的圆。扩展到树上的情形时，也希望得到类似结论。

本节将探索的主要问题是：对于树上的两个点集 S, T，$\mathcal{K}(S), \mathcal{K}(T)$ 和 $\mathcal{K}(S \cup T)$ 之间的关系。换句话说，若只知道 $\mathcal{K}(S), \mathcal{K}(T)$，能否在不知道 S, T 本身为何值时，快速求出 $\mathcal{K}(S \cup T)$。

考虑对 $\mathcal{K}(S)$ 和 $\mathcal{K}(T)$ 的包含关系进行讨论：

- 若 $\mathcal{K}(S) \subset \mathcal{K}(T)$，即 $dis(m(S), m(T)) + \frac{d(S)}{2} \leq \frac{d(T)}{2}$，则显然 $\mathcal{K}(S \cup T) = \mathcal{K}(T)$。

- 若 $\mathcal{K}(T) \subset \mathcal{K}(S)$，即 $dis(m(S), m(T)) + \frac{d(T)}{2} \leq \frac{d(S)}{2}$，则显然 $\mathcal{K}(S \cup T) = \mathcal{K}(S)$。

- 否则有 $\mathcal{K}(S) \not\subset \mathcal{K}(T)$ 且 $\mathcal{K}(T) \not\subset \mathcal{K}(S)$，即两者互不包含。结论为：

$$\mathcal{K}(S \cup T) = N\left(\frown\left(m(S), m(T), \frac{len - d(S)/2 + d(T)/2}{2}\right), \frac{len + d(S)/2 + d(T)/2}{2}\right)$$

  其中 $len = dis(m(S), m(T))$。

  尝试证明上述结论。对于点集 S，必然能找到直径 $\mathrm{Path}(u_1, v_1)$，不妨设 u_1 不与 $m(T)$ 在 $m(S)$ 的相同子树中；同理能在 T 中找到直径 $\mathrm{Path}(u_2, v_2)$，u_2 不与 $m(S)$ 在 $m(T)$ 的相同

子树中。则 $dis(u_1, u_2) = dis(u_1, m(S)) + dis(m(S), m(T)) + dis(m(T), u_2) = len + d(S)/2 + d(T)/2$，该值一定是 $S \cup T$ 的直径长度。

从而这条直径的中点就是 $\curvearrowright(u_1, u_2, \frac{len+d(S)/2+d(T)/2}{2}) = \curvearrowright(m(S), m(T), \frac{len-d(S)/2+d(T)/2}{2})$。

通过上述讨论可知，在只知道 $\mathcal{K}(S)$ 和 $\mathcal{K}(T)$ 的情况下就可以直接求出 $\mathcal{K}(S \cup T)$，该结论也通常被称为树上圆理论。拥有该结论后，若需要维护点集的直径并支持合并，可以不维护整个点集 S，而是只维护 $\mathcal{K}(S)$。

该种维护方式相较于传统的维护点集内任意一条直径更加简洁，在下面的例题中很容易看出。

例题 1 (Range Diameter Sum[1]). 给定一棵包含 n 个点的树，定义：

$$\mathrm{diam}(l, r) = \max_{l \le u, v \le r} dis(u, v)$$

计算：

$$\sum_{1 \le l \le r \le n} \mathrm{diam}(l, r)$$

数据范围：$1 \le n \le 10^5$。

解法　求单个 $\mathrm{diam}(l, r)$ 即为树上邻域的标准形式。现在需要求解所有子区间的答案之和，求解类似问题的经典做法是使用分治。

对于一个分治区间 $[l, r]$，令 $mid = \lfloor \frac{l+r}{2} \rfloor$，递归分治求解 $[l, mid]$ 和 $[mid + 1, r]$ 两个区间的所有子区间答案，而在当前分治过程中计算跨过 mid 的区间答案。

预处理出所有形如 $\mathcal{K}([x, mid])$ 和 $\mathcal{K}([mid+1, y])$ 的树上圆，接下来即求这些树上圆两两并半径之和。

固定 x，则 $\mathcal{K}([x, mid])$ 被确定。y 从 $mid + 1$ 增加到 y 的过程中，$\mathcal{K}([mid + 1, y])$ 逐渐增大，因此 $\mathcal{K}([mid + 1, y])$ 和 $\mathcal{K}([x, mid])$ 的关系也可以被分为三段：

- 第一段中，$\mathcal{K}([x, mid]) \supset \mathcal{K}([mid + 1, y])$，此时贡献即为 $\mathcal{K}([x, mid])$ 的直径。
- 第二段中，两个树上圆互不包含。根据树上圆求并的讨论，两个树上圆求并后新树上圆的直径是 $dis(m([x, mid]), m([mid + 1, r])) + d([x, mid])/2 + d([mid + 1, y])/2$（此处分别用 $m(S)$ 和 $d(S)$ 来表示 $\mathcal{K}(S)$ 的中点和直径）。
- 第三段中，$\mathcal{K}([x, mid]) \subset \mathcal{K}([mid + 1, y])$，此时贡献即为 $\mathcal{K}([mid + 1, y])$。

对于第一段和第三段，贡献都是容易处理的。对于第二段 $d([x, mid]) + d([mid + 1, y])$ 部分也容易计算。而 $dis(m[x, mid], m[mid + 1, y])$ 部分可以转化为以下问题：动态维护一个可重点集 S，需要支持以下两种操作：

[1]题目来源：Codeforces Round 691 (Div. 1) F. Range Diameter Sum https://codeforces.com/contest/1458/problem/F.

- 向 S 中插入或删除顶点。
- 给定顶点 x，查询 $\sum_{y\in S} dis(x,y)$。

该问题可以使用点分树或重链剖分解决。视最后一步的具体维护与实现方式，复杂度在 $O(n\log^2 n)$ 到 $O(n\log^3 n)$ 不等。 ■

通过上述例题可以发现，树上圆理论能够让点集之间的包含关系刻画更加具体。

2.3 树上邻域求交

本小节中将主要讨论树上真实邻域求交的问题。经过分析得到的最终结论为：树上任意两个真实邻域求交的结果必为真实邻域或空集。接下来将对该结论进行证明。

定理 2.3. 树上任意两个真实邻域 $N(u_1,r_1)$ 和 $N(u_2,r_2)$ 的交必为真实邻域或空集。

证明. 容易观察到两个真实邻域之交为空当且仅当 $dis(u_1,u_2) > r_1 + r_2$。否则必有：

$$N(u_1,r_1)\cap N(u_2,r_2) = N\left(\frown\left(u_1,u_2,\frac{r_1-r_2+dis(u_1,u_2)}{2}\right),\frac{r_1+r_2-dis(u_1,u_2)}{2}\right)$$

令 $len = dis(u_1,u_2)$，广义点 $c = \frown(u_1,u_2,\frac{r_1-r_2+len}{2})$。根据 $\frown$ 的性质必然有 $dis(u_1,c) = \frac{r_1-r_2+len}{2}$ 和 $dis(u_2,c) = len - \frac{r_1-r_2+len}{2} = \frac{r_2-r_1+len}{2}$。则对于任意 $t\in N(c,\frac{r_1+r_2-len}{2})$，都有：

$$\begin{cases} dis(t,u_1)\le dis(t,c)+dis(c,u_1)\le \frac{r_1+r_2-len}{2}+\frac{r_1-r_2+len}{2} = r_1 & \Longrightarrow\ t\in N(u_1,r_1)\\ dis(t,u_2)\le dis(t,c)+dis(c,u_2)\le \frac{r_1+r_2-len}{2}+\frac{r_2-r_1+len}{2} = r_2 & \Longrightarrow\ t\in N(u_2,r_2)\end{cases}$$

由此可知 $N(c,\frac{r_1+r_2-len}{2})\subset(N(u_1,r_1)\cap N(u_2,r_2))$。

再证明对于任意一点 t，若 $t\in N(u_1,r_1)$ 且 $t\in N(u_2,r_2)$，则必有 $t\in N(c,\frac{r_1+r_2-len}{2})$。反正，存在一个点使得 $t\notin N(c,\frac{r_1+r_2-len}{2})$，那么有 $dis(t,c) > \frac{r_1+r_2-len}{2}$。由于 u_1,u_2 一定在 c 的不同两棵子树中，因此以下两个条件之一必有一者成立：$dis(t,u_1) = dis(t,c)+dis(c,u_1)$ 或 $dis(t,u_2) = dis(t,c)+dis(c,u_2)$。不妨设成立前者，那么 $dis(t,u_1) = dis(t,c)+dis(c,u_1) > \frac{r_1+r_2-len}{2}+\frac{r_1-r_2+len}{2} = r_1$，这与 $t\in N(u_1,r_1)$ 矛盾。因此 $N(c,\frac{r_1+r_2-len}{2})\supset(N(u_1,r_1)\cap N(u_2,r_2))$

结合上述两个性质可知：$N(u_1,r_1)\cap N(u_2,r_2) = N(c,\frac{r_1+r_2-len}{2})$。 □

若视空集也是树上真实邻域，则由上述分析可以发现：树上真实邻域对求交运算封闭。

但该结论仍然不够优美，因为树上真实邻域的中心点可能并不是树上的节点。在下一节中，将引入树上链邻域的概念，从而让邻域求交的表示更加方便，不需要再使用广义点来作为邻域中点。

3 树上链邻域理论

3.1 链邻域与邻域的关系

在“记号与约定”一节中已经提及，本部分所有的链邻域均不包含广义点。也就是说，所有 $M(u, v, r)$ 中 u, v 均为真实的树上顶点，r 一定是整数。

上一节中已经讨论了树上邻域求交的相关问题。链邻域则是邻域的一种扩展形式，他能够表示所有真实的邻域。

引理 3.1 (真实邻域的链邻域表示)**.** 任意一个真实邻域都可以用链邻域来表示。

证明. 对任意一个真实邻域 $N(u, r)$，分 u 的类型进行讨论：

- 对于 $u \in V$ 的真实邻域 $N(u, r)$，可以直接用链邻域 $M(u, u, r)$ 来表示。
- 对于 $u \notin V$ 的真实邻域 $N(u = \mathcal{M}(p, q), r)$，可以用链邻域 $M(p, q, r - \frac{1}{2})$ 来表示。

综上，无论 u 为何种类型，均可以用一个链邻域来表示该真实邻域。 □

本节中，将最终证明链邻域求交之后必然是链邻域或空集。

3.2 零半径树上链邻域求交

该问题亦可描述为树上两条链求交。

定理 3.1 (链求交定理)**.** 对于树上任意两条链 $\text{Path}(u_1, v_1), \text{Path}(u_2, v_2)$，他们的交必为一条链或空集。

证明. 该定理是经典结论。首先他们的交必然是 $\text{Path}(u_1, v_1)$ 的子集，故必为若干段链。若段数 ≥ 2，则根据这些链也是 $\text{Path}(u_2, v_2)$ 的子集可以推出 $T = (V, E)$ 中包含环，与树的定义矛盾。

故其交集必为一条链或零条链（即空集）。 □

3.3 等半径树上链邻域求交

本小节将证明对于任意两个半径相等的链邻域 $M(u_1, v_1, r)$ 和 $M(u_2, v_2, r)$，求交后仍然为链邻域或空集，并给出一个较为简单的表达形式。

对于树上的任意两条链 $\text{Path}(u_1, v_1)$ 和 $\text{Path}(u_2, v_2)$，他们之间的位置关系可以分为两种：相交和不相交。对于这两种情况分别进行讨论。

引理 3.2. 对于树上两条相交的链 $\mathrm{Path}(u_1, v_1), \mathrm{Path}(u_2, v_2)$ 和任意整数 $r \geq 0$，设 $\mathrm{Path}(x, y) = \mathrm{Path}(u_1, v_1) \cap \mathrm{Path}(u_2, v_2)$，则 $M(u_1, v_1, r) \cap M(u_2, v_2, r) = M(x, y, r)$。

证明. 不妨设 u_1, u_2 在 x 侧（即 $dis(u_1, x) \leq dis(u_1, y)$，其他同理），$v_1, v_2$ 在 y 侧。

若 $t \in M(x, y, r)$，则根据定义必然存在一点 $c \in \mathrm{Path}(x, y)$ 满足 $dis(c, t) \leq r$。而显然 $c \in \mathrm{Path}(u_1, v_1), \mathrm{Path}(u_2, v_2)$，因此也有 $t \in M(u_1, v_1, r), M(u_2, v_2, r)$。这说明 $M(x, y, r) \subset (M(u_1, v_1, r) \cap M(u_2, v_2, r))$。

同时，如果一个点 t 同时满足 $t \in M(u_1, v_1, r)$ 和 $t \in M(u_2, v_2, r)$，假设 t 属于 $\mathrm{Path}(u_1, v_1)$ 中的 c。则分两种情况讨论：

1. 若 $c \in \mathrm{Path}(x, y)$，那么由 $t \in M(u_1, v_1, r)$ 可知 $dis(t, c) \leq r$，因此也有 $t \in M(x, y, r)$。
2. 否则必有 $c \in \mathrm{Path}(x, u_1)$ 或 $c \in \mathrm{Path}(v_1, y)$ 之一成立。不妨设前者成立。则此时有 t 属于 $\mathrm{Path}(u_2, v_2)$ 中的 x，由 $t \in M(u_2, v_2, r)$ 可知 $dis(t, x) \leq r$，因此也有 $t \in M(x, y, r)$。

故无论 c 在何处，必有 $t \in M(x, y, r)$，也即 $(M(u_1, v_1, r) \cap M(u_2, v_2, r)) \subset M(x, y, r)$。

结合两个包含关系可得：$M(u_1, v_1, r) \cap M(u_2, v_2, r) = M(x, y, r)$。 □

引理 3.3. 对于树上两条不相交的链 $\mathrm{Path}(u_1, v_1), \mathrm{Path}(u_2, v_2)$ 和任意整数 $r \geq 0$，$M(u_1, v_1, r) \cap M(u_2, v_2, r)$ 必为一个真实邻域或空集。

证明. 不妨设 x 为 $\min\limits_{p \in \mathrm{Path}(u_1, v_1)} dis(p, \mathrm{Path}(u_2, v_2))$ 取到最小值时的 p，y 为 $\min\limits_{q \in \mathrm{Path}(u_2, v_2)} dis(q, \mathrm{Path}(u_1, v_1))$ 取到最小值时的 q。容易证明 x, y 都是唯一的。又令 $len = dis(x, y)$。

若 $len > 2r$，则显然两个链邻域无交，交集为空集。

否则必有 $len \leq 2r$。令广义点 $c = \curvearrowright (x, y, \frac{len}{2})$。

此时，发现对于所有 $dis(t, y) = dis(t, c) + dis(c, y)$ 的点 t（形象地说，即为在 c 靠近 x 一侧的点），都有 $dis(t, y) = dis(t, c) + dis(c, y) = dis(t, c) + dis(c, x) \geq dis(t, x)$，因此对于这一些点可以忽略 $M(u_1, v_1, r)$ 的限制，只需要满足 $dis(t, y) \leq r$ 即 $dis(t, c) + dis(c, y) \leq r$ 即 $dis(t, c) \leq r - dis(c, y) = r - \frac{len}{2}$ 即可。

同理，对于所有 $dis(t, x) = dis(t, c) + dis(c, x)$ 的点也都只需要满足 $dis(t, c) \leq r - \frac{len}{2}$ 即可。所有满足这两个条件之一的点构成了所有点的全集。故两个链邻域的交集即为 $N(c, r - \frac{len}{2})$。 □

结合引理 3.2，引理 3.3 和定理 3.1 可得到如下定理：

定理 3.2 (等半径链邻域求交定理)**.** 对于任意两个半径相等的树上链邻域 $M(u_1, v_1, r)$ 和 $M(u_2, v_2, r)$，$M(u_1, v_1, r) \cap M(u_2, v_2, r)$ 必为空集，或可以被表示为链邻域的形式。

3.4 一般树上链邻域求交

上一小节中证明了等半径的两个链邻域相交仍是链邻域或空集。此小节将讨论半径不同时的情况。经过分析发现此性质仍然满足。下面将逐步对该性质进行证明。

仍然考虑两个链邻域的中心链的位置关心进行分类讨论。

引理 3.4. 对于树上两条相交的链 $\mathrm{Path}(u_1, v_1), \mathrm{Path}(u_2, v_2)$ 和任意两个整数 $r_1, r_2 \geq 0$，有 $M(u_1, v_1, r_1) \cap M(u_2, v_2, r_2)$ 仍为链邻域。

证明. 不妨设 $r_1 \leq r_2$，u_1, v_1 在 x 侧，u_2, v_2 在 y 侧。

首先对整棵树进行一些变换：如果 $dis(u_1, x) < r_2 - r_1$，则在 u_1 节点上接一条长度为 $r_2 - r_1 - dis(u_1, x)$ 的虚链，同时令 u_1' 为虚链的链底；否则令 $u_1' = u_1$。对 v_1 进行类似操作得到 v_1'。

现在必有 $dis(u_1', x), dis(v_1', y) \geq r_2 - r_1$ 成立。考虑先求 $M(u_1', v_1', r_1)$ 和 $M(u_2, v_2, r_2)$ 的交。注意此时新添加的虚链并不影响两条链的交。

接下来说明下述等式成立：

$$M(u_1', v_1', r_1) \cap M(u_2, v_2, r_2) = M(u_1', v_1', r_1) \cap M(\frown(x, u_1', r_2 - r_1), \frown(y, v_1', r_2 - r_1), r_1)$$

令 $x' = \frown(x, u_1', r_2 - r_1), y' = \frown(y, u_2', r_2 - r_1)$。

上述添加虚链的过程保证了 $dis(u_1', x), dis(v_1', y) \geq r_2 - r_1$，因此 x' 和 y' 必然是存在的。

首先说明 $\forall t \in M(x', y', r_1)$，都有 $t \in M(u_2, v_2, r_2)$。设 t 在 $\mathrm{Path}(x', y')$ 链上属于点 c。

1. 若 $c \in \mathrm{Path}(x, y)$,则由 $\mathrm{Path}(x, y) \subset \mathrm{Path}(u_2, v_2)$ 可知 $dis(t, \mathrm{Path}(u_2, v_2)) \leq dis(t, \mathrm{Path}(x, y)) \leq r_1 \leq r_2$，显然 $t \in M(u_2, v_2, r_2)$。

2. 若 $c \notin \mathrm{Path}(x, y)$，则 $c \in \mathrm{Path}(x', x) \setminus \{x\}$ 和 $c \in \mathrm{Path}(y', y) \setminus \{y\}$ 之一必然成立。不妨设成立前者，则有 $dis(t, x) = dis(t, c) + dis(c, x) \leq r_1 + (r_2 - r_1) = r_2$，因此 $t \in M(u_2, v_2, r_2)$。

由此可知 $M(x', y', r_1) \subset M(u_2, v_2, r_2)$，故有：

$$(M(u_1', v_1', r_1) \cap M(u_2, v_2, r_2)) \supset (M(u_1', v_1', r_1) \cap M(x', y', r_1))$$

接下来说明对于任意顶点 t，若 $t \in M(u_1', v_1', r_1)$，$t \notin M(x', y', r_1)$，则必有 $t \notin M(u_2, v_2, r_2)$。

考虑反证。设存在这样顶点 t 同时满足下面三个条件：$t \in M(u_1', v_1', r_1)$，$t \notin M(x', y', r_1)$，$t \in M(u_2, v_2, r_2)$。

设 t 属于 $\mathrm{Path}(u_1', v_1')$ 的 c，由于 $t \notin M(x', y', r_1)$，因此必有 $c \notin \mathrm{Path}(x', y')$，即下列两个条件之一成立：$c \in \mathrm{Path}(u_1', x') \setminus \{x'\}$ 或 $c \in \mathrm{Path}(v_1', y') \setminus \{y'\}$。不妨设成立前者。由 $t \notin M(x', y', r_1)$ 可得 $dis(t, x') > r_1$。

当以 x 为根时，由于 $t \in \mathrm{Path}(u_1', x') \setminus \{x'\}$，因此 t 一定在 x' 的子树内，进一步得到 t 在链 $\mathrm{Path}(u_2, v_2)$ 上属于 x。故 $dis(t, \mathrm{Path}(u_2, v_2)) = dis(t, x) = dis(t, x') + dis(x', x) > r_1 + (r_2 - r_1) = r_2$，与 $t \in M(u_2, v_2, r_2)$ 矛盾，假设不成立。

因此得到若 $t \in M(u_1', v_1', r_1)$，$t \notin M(x', y', r_1)$，则必有 $t \notin M(u_2, v_2, r_2)$，也就是说：

$$(M(u_1', v_1', r_1) \cap M(u_2, v_2, r_2)) \subset (M(u_1', v_1', r_1) \cap M(x', y', r_1))$$

结合两个结论就可以得到成立：

$$M(u_1', v_1', r_1) \cap M(u_2, v_2, r_2) = M(u_1', v_1', r_1) \cap M(\curvearrowright(x, u_1', r_2 - r_1), \curvearrowright(y, v_1', r_2 - r_1), r_1)$$

同时根据加虚链的方式容易知道 $M(u_1, v_1, r_1) \subset M(u_1', v_1', r_1')$，因此：

$$\begin{aligned}
&M(u_1, v_1, r_1) \cap M(u_2, v_2, r_2)\\
=&M(u_1, v_1, r_1) \cap M(u_1', v_1', r_1) \cap M(u_2, v_2, r_2)\\
=&M(u_1, v_1, r_1) \cap M(u_1', v_1', r_1) \cap M(\curvearrowright(x, u_1', r_2 - r_1), \curvearrowright(y, v_1', r_2 - r_1), r_1)\\
=&M(u_1, v_1, r_1) \cap M(\curvearrowright(x, u_1', r_2 - r_1), \curvearrowright(y, v_1', r_2 - r_1), r_1)
\end{aligned}$$

由上式可知该交集其实是两个等半径链邻域求交的结果，由定理 3.2 可知结果必然是链邻域或空集。

又由 $\text{Path}(x, y) \subset M(u_1, v_1, r_1) \cap M(u_2, v_2, r_2)$ 可知结果非空，故结果必然为链邻域的形式。 □

引理 3.5. 对于树上两条不交的链 $\text{Path}(u_1, v_1), \text{Path}(u_2, v_2)$ 和任意两个整数 $r_1, r_2 \geq 0$，有 $M(u_1, v_1, r_1) \cap M(u_2, v_2, r_2)$ 必为一个链邻域或空集。

证明. 与等半径的证明方式类似，仍然不妨设 x 为 $\min\limits_{p \in \text{Path}(u_1, v_1)} dis(p, \text{Path}(u_2, v_2))$ 取到最小值时的 p，y 为 $\min\limits_{q \in \text{Path}(u_2, v_2)} dis(q, \text{Path}(u_1, v_1))$ 取到最小值时的 q。容易证明 x, y 都是唯一的。又令 $len = dis(x, y)$。

若 $len > r_1 + r_2$ 则显然交集为空集。否则令 $w = \frac{len + r_1 - r_2}{2}$。

- 如果 $0 \leq w \leq len$，那么可以找到一广义点 $c = \curvearrowright(x, y, w)$。

  发现对于一个点 c 在两个链邻域的交中的等价条件为 $r_1 - dis(c, \text{Path}(u_1, v_1)) \geq 0$ 且 $r_2 - dis(c, \text{Path}(u_2, v_2)) \geq 0$。

  此时，对于所有满足 $dis(t, y) = dis(t, c) + dis(c, y)$ 的点 t（形象的看为 c 靠 x 一侧的点），都有 $r_1 - dis(t, \text{Path}(u_1, v_1)) \geq r_1 - dis(t, x) \geq r_1 - dis(t, c) - dis(c, x) = r_1 - dis(t, c) - w = r_2 - len + w - dis(t, c) = r_2 - len(c, y) - dis(t, c) = r_2 - dis(t, y) = r_2 - dis(t, \text{Path}(u_2, v_2))$，即对于这一部分的点只需要考虑 $r_2 - dis(t, \text{Path}(u_2, v_2)) \geq 0$ 的限制即可，该限制又可被改写为 $r_2 - dis(t, c) - (len - w) \geq 0$ 即 $dis(t, c) \leq r_2 - len + w = r_1 - w$。

  对于另一边满足 $dis(t, x) = dis(t, c) + dis(c, x)$ 的点 t（即 c 靠 y 一侧的点）类似的也可将限制化简为 $dis(t, c) \leq r_1 - w$。发现这样两个点集的并集恰好为所有顶点全集。故此时满足限制的点即为 $N(c, r_1 - w)$。且发现 $N(c, r_1 - w)$ 必然是真实邻域，再由定理 3.1 即可得到结果能由链邻域的形式表示。

- 否则有 $w < 0$ 和 $w > len$ 两者之一成立。两者分别等价于 $r_2 > r_1 + len$ 于 $r_1 > r_2 + len$。不妨设成立前者，即有 $r_2 > r_1 + len$。

  接下来将说明可以将 $M(u_2, v_2, r_2)$ 的限制放缩为 $N(x, r_2 - len)$。

  首先证明 $N(x, r_2 - len) \subset M(u_2, v_2, r_2)$。若一个点 $t \in N(x, r_2 - len)$，则 $dis(t, x) \le r_2 - len$，则有 $dis(t, \text{Path}(u_2, v_2)) \le dis(t, y) \le dis(t, x) + dis(x, y) \le (r_2 - len) + len = r_2$，故 $t \in M(u_2, v_2, r_2)$。进而可得 $(M(u_1, v_1, r_1) \cap N(x, r_2 - len)) \subset (M(u_1, v_1, r_1) \cap M(u_2, v_2, r_2))$。

  若一个点 $t \in M(u_1, v_1, r_1)$ 满足 $t \notin N(x, r_2 - len)$，则可以得到 $dis(t, x) > r_2 - len > r1$，但是 $dis(t, \text{Path}(u_1, v_1)) \le r_1$，因此 t 一定属于 $\text{Path}(u_1, v_1)$ 链上的一个非 x 的点。那么就有 $dis(t, \text{Path}(u_2, v_2)) = dis(t, y) = dis(t, x) + dis(x, y) > (r_2 - len) + len = r_2$，即 $t \notin M(u_2, v_2, r_2)$。进而有 $(M(u_1, v_1, r_1) \cap N(x, r_2 - len)) \supset (M(u_1, v_1, r_1) \cap M(u_2, v_2, r_2))$。

  结合上述两个条件就有 $(M(u_1, v_1, r_1) \cap N(x, r_2 - len)) = (M(u_1, v_1, r_1) \cap M(u_2, v_2, r_2))$。因此这两个链邻域求交的结果就是 $M(u_1, v_1, r_1)$ 和 $N(x, r_2 - len) = M(x, x, r_2 - len)$ 求交的结果，而 $x \in \text{Path}(u_1, v_1)$，因此将问题转化为了两个中心链有交的链邻域求交问题，由引理 3.4 可知，结果必然为链邻域的形式。

综上，无论 w 为何值，均有结果为链邻域或空集。引理得证。 □

结合引理 3.4 和引理 3.5 可得到如下具有一般性的且优美的定理：

定理 3.3 (链邻域求交定理). 对于任意两个树上链邻域 $M(u_1, v_1, r_1)$ 和 $M(u_2, v_2, r_2)$，都有 $M(u_1, v_1, r_1) \cap M(u_2, v_2, r_2)$ 必为空集，或可以被表示为链邻域的形式。

也就是说，若将空集也视作链邻域，则一棵树的链邻域对求交操作封闭。

3.5 算法与例题

如果将空集也视为一种特殊的链邻域，那么容易使用一个结构体来存储链邻域。

当需要对两个链邻域求交时，算法的主体思想在上一小节的证明过程中已经给出。除去所有较为复杂的分类讨论以外，主要需要进行以下几种操作：

1. 给出树上的两个点 u, v，查询 $dis(u, v)$。

2. 给出树上的一条链 $\text{Path}(u, v)$ 和一点 x，查询 $dis(x, \text{Path}(u, v))$ 以及 x 属于链 $\text{Path}(u, v)$ 上的哪一点。

3. 给出树上的两个点 u, v 和一个非负整数 $d \le dis(u, v)$，查询 $\frown (u, v, d)$。

上述三种操作均为树上问题的经典操作，存在多种不同解法。设 $n = |V|$，下面将给出这三个问题的一种较为简单的 $O(n)$ 预处理，$O(\log n)$ 进行单次查询的办法：

1. 任选一节点作为根节点，对整棵树进行重链剖分。

2. 由重链剖分经典问题的求解方法，容易单次 $O(\log n)$ 查询任意两点的最近公共祖先，以及一个点的 k 级祖先。定义两点 u, v 的最近公共祖先为 $\mathrm{LCA}(u, v)$，一个点 u 的 k 级祖先为 $\Uparrow_k (u)$。

3. 对于操作 1，答案即为 $dep_u + dep_v - 2 \cdot dep_{\mathrm{LCA}(u,v)}$。

4. 对于操作 2，逐个求解 $A = \mathrm{LCA}(u, v), B = \mathrm{LCA}(u, x), C = \mathrm{LCA}(v, x)$，设 p 为 A, B, C 三点中深度最大的节点，则 x 必属于链 $\mathrm{Path}(u, v)$ 上的 p 点。再求 $dis(x, p)$ 即可得到 $dis(x, \mathrm{Path}(u, v))$。

5. 对于操作 3，首先求出 $w = \mathrm{LCA}(u, v)$。如果 $dis(u, w) \geq d$，则 $\curvearrowright (u, v, d) = \Uparrow_d (u)$；否则 $\curvearrowright (u, v, d) = \Uparrow_{dis(u,v)-d} (v)$。

由上一小节中讨论的过程可知，所有的链邻域求交操作都可以在进行常数次上述三种操作来完成。因此，可以做到 $O(n)$ 对树进行预处理，随后在 $O(\log n)$ 的复杂度内完成一次链邻域求交操作。

接下来将给出几道与链邻域求交有关的例题进行分析。

例题 2 (Ald[2])**.** 给定一棵 n 个顶点构成的树，顶点从 $1 \sim n$ 标号。你需要维护一个由树上路径构成的可重集 S，并支持以下几种操作共 q 次：

1. 给定两个顶点 u, v，将 $\mathrm{Path}(u, v)$ 加入到可重集 S 中。

2. 给定两个顶点 u, v，将一个 $\mathrm{Path}(u, v)$ 从可重集 S 中删除。

3. 给定一个非负整数 d，求所有以 S 中路径为中心链的 $d-$ 链邻域的交的大小，即：

$$\left| \bigcap_{\mathrm{Path}(u,v) \in S} M(u, v, d) \right|$$

数据范围：$1 \leq n, q \leq 10^5$。

解法　根据定理 3.2 的结论，$\bigcap_{\mathrm{Path}(u,v)\in S} M(u, v, d)$ 一定可以被表示为链邻域的形式。链邻域数点问题在《浅谈一类树上统计相关问题》[1] 一文中进行了详细介绍。这一部分并不是本文重点，读者可以自行思考或阅读该篇论文。

接下来将讨论如何求出每一次 3 类型询问时交集的链邻域表示，该问题也是本例题中与本文关联最大的部分。

[2]题目来源：Petrozavodsk Summer 2023. Day 7. PKU Contest G. Ald. https://qoj.ac/contest/1376/problem/7507.

根据引理 3.2 的结论，对两个中心链相交的链邻域求交，结果就是半径相等，中心链为原本两条中心链交部分的链邻域。

将此结论推广到任意有限集合中依然成立。即：若 $\bigcap_{\mathrm{Path}(u,v)\in S} \mathrm{Path}(u,v) \neq \varnothing$，那么下式成立：

$$\bigcap_{\mathrm{Path}(u,v)\in S} M(u,v,d) = M(u',v',d). \quad \text{where } \mathrm{Path}(u',v') = \bigcap_{\mathrm{Path}(u,v)\in S} \mathrm{Path}(u,v)$$

否则该结论并不成立。但是有如下结论成立：

引理 3.6. 对于由链构成的集合 S 满足 $\bigcap_{\mathrm{Path}(u,v)\in S} \mathrm{Path}(u,v) = \varnothing$，对于任意两条 S 中的链 $\mathrm{Path}(u_1,v_1),\mathrm{Path}(u_2,v_2) \in S$ 满足 $dis(\mathrm{Path}(u_1,v_1),\mathrm{Path}(u_2,v_2)) = \max_{p_1,p_2\in S} dis(p_1,p_2)$，则成立下式：

$$\bigcap_{\mathrm{Path}(u,v)\in S} M(u,v,d) = M(u_1,v_1,d) \cap M(u_2,v_2,d)$$

证明. 由于所有中心链的交为空集，因此必然存在两条链的交为空，从而距离最长的两条中心链交为空，即有 $dis(\mathrm{Path}(u_1,v_1),\mathrm{Path}(u_2,v_2)) > 0$。

如果 $M(u_1,v_1,d) \cap M(u_2,v_2,d) = \varnothing$，那么显然等式左右均为空集，等式成立。

否则根据引理 3.3，设 $x \in \mathrm{Path}(u_1,v_1), y \in \mathrm{Path}(u_2,v_2)$ 为让两条链距离取到最小值的 x,y，令 $len = dis(x,y)$，$c = \curvearrowright (x,y,\frac{len}{2})$，那么 $M(u_1,v_1,d) \cap M(u_2,v_2,d) = N(\curvearrowright (x,y,\frac{len}{2}), d - \frac{len}{2})$。

由 $\mathrm{Path}(u_1,v_1)$ 和 $\mathrm{Path}(u_2,v_2)$ 是所有链对中距离最大的可知，对于任意 $\mathrm{Path}(u',v') \in S$，都有 $dis(c,\mathrm{Path}(u',v')) \leq \frac{len}{2}$。设 c 属于链 $\mathrm{Path}(u',v')$ 中的 t，则有 $N(\curvearrowright (x,y,\frac{len}{2}), d - \frac{len}{2}) \subset N(t,d) \subset M(u',v',d)$。引理得证。 □

利用引理 3.6 的结论，只需要维护出 S 集合中距离最远的两条链即可。

得到上述结论后，使用线段树分治将问题转化为只向 S 集合中添加路径。同时维护以下两个信息：

- 若 S 中所有路径交集不为空，则维护所有路径的交集。
- 否则维护 S 中距离最大的两条链。

不难发现上述信息在新增加一条路径时都可以在 $O(\log n)$ 的复杂度内进行更新。因此本题可以在 $O(n + q\log q\log n)$ 的复杂度内求出每一个 3 询问对应的链邻域表示。 ■

例题 3 (这里有只毛毛虫[3]). 给定一张 n 个点 n 条边构成的无向连通图（即基环树）。

定义基环树上任意两个节点的距离 $dis(u,v)$ 为 u 与 v 之间最短路径的边数。

定义一个该基环树上的一个毛毛虫 Cat(u,v,c) 为：

$$\mathrm{Cat}(u,v,c) = \{x \mid dis(x,u) + dis(x,v) \leq dis(u,v) + c\}$$

你需要维护一个由毛毛虫构成的可重集 S，支持以下操作 q 次：

[3]本题为笔者原创题。

- 给定三个整数 u, v, c，将 $\text{Cat}(u, v, c)$ 插入集合 S，并在插入后输出 S 中所有毛毛虫交集的大小。

数据范围：$1 \le n, q \le 5 \times 10^5$。

解法　若本题的无向图是一棵树，则该问题即为上面所介绍的树上链邻域求交。

考虑求出基环树的环，将所有环上的边切断后，原图变成若干棵树，每棵树都以一个环上的点为根。尝试对于每一棵拆出来的树维护对应树上的交。

考虑一次操作会对每个树产生什么影响。假设操作为 $\text{Cat}(u, v, c)$，则对于 u 和 v 所在的树，影响即为在对应的树上进行一次链邻域求交操作。

而对于非 u 和 v 所在的树，其在 $\text{Cat}(u, v, c)$ 的部分其实是一个根的邻域。具体的，设根为 t，则 t 树内与 $\text{Cat}(u, v, c)$ 相交的部分即为 $N(t, c - dis(c, \text{Path}(u, v)))$。

通过这一部分分析可以发现，对于每一棵树，每次操作本质上是进行一次链邻域求教操作（由定理 3.1，邻域求交也可以被表示为链邻域求交）。故每一棵树中在 S 交中的部分即为树上的一个链邻域或空。

接下来尝试快速维护这个过程。对于一棵以 t 为根的树的链邻域交，容易被表示为 $M(u_t, v_t, r_t) \cap N(t, d_t)$，初始时 d_t 为对应树中最深节点的深度。一次操作中，会对 $M(u_t, v_t, r_t)$ 产生影响的只有 u 和 v 所在的树中，直接暴力更新即可。

而对于别的树，只会对 $N(t, d_t)$ 这一部分中的 d_t 产生影响，且具体影响为将 d_t 对一个值取较小值。尝试快速找到所有 d_t 减小的 t，并对这样的 d_t 重新求链邻域之交。容易发现这样的更新只会进行 $\sum dep_t = O(n)$ 次。而对 d_t 取较小值的值一定形如常数段公差为 0 或 1 的等差数列，因此可以直接用线段树维护环上每个节点对应树当前的 d_t、$d_t + rnk_t$ 和 $d_t - rnk_t$ 即可快速找出一次操作会更新的位置。

综上，本题只需要进行 $O(n + q)$ 次链邻域求交操作。而对于每一次求交操作后，还需要进行一次链邻域数点操作，该操作可以参考《浅谈一类树上统计相关问题》[1] 一文中所给出的做法。至此，本题可以在 $O((n + q) \log n)$ 的时间复杂度内被解决。 ■

4 总结

本文从树上邻域出发，首先讨论了树上邻域与点集直径之间的关系，并发现树上真实邻域对求交操作封闭。进一步地，将树上邻域推广到树上链邻域，通过证明若干引理，发现若将空集也视作树上链邻域，则树上链邻域对求交操作也封闭。同时，也给出了实现树上链邻域求交的具体算法实现方法，并给出几道例题加以分析，对此操作的应用进行了初步的展示。

目前本理论与算法在信息学竞赛的具体题目中出现与应用较少。希望本文能起到抛砖引玉的作用，期待读者能够对部分理论进行进一步研究。

5 致谢

感谢中国计算机学会提供学习和交流的平台。
感谢杭州第二中学李建老师的关心和指导。
感谢家人、朋友对我的支持与鼓励。
感谢陈昕阳、孙恒喆学长和章绍嘉同学对我的启发。
感谢其他给予我帮助的老师与同学。

参考文献

[1] 朱羿恺. (2023). 浅谈一类树上统计相关问题. 2023 年信息学奥林匹克中国国家集训队论文.

[2] 王羽立. (2020). 题解 CF1458F【Range Diameter Sum】. Luogu. https://www.luogu.com.cn/article/rb0wgud8.

[3] 吴思成. (2023). 冬のエピローグ. Luogu. https://www.luogu.com.cn/article/hop4knd8.

[4] 朱羿恺. (2023). Petrozavodsk Summer 2023. Day 7. PKU Contest Tutorials. QOJ. https://qoj.ac/download.php?type=attachments&id=1376&r=1.