浅谈树上邻域与链邻域的并交问题

Union and Intersection of Tree and Path Neighborhoods

汪苏轶

杭州第二中学

2026 年 2 月

广义点与广义距离

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。

广义点与广义距离

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

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

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 $\mathrm{Path}(s,t)$，定义 u 到 $\mathrm{Path}(s,t)$ 的距离：

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

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

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（$0 \le \theta \le d_T(s, t)$）。

形式化记作：

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

树上邻域与真实邻域

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

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

称 u 为邻域 $N(u,r)$ 的中心，r 为该邻域的半径。

树上邻域与真实邻域

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

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

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。

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 $\leq r$ 的点所构成的集合，记作 $M(u, v, r)$。形式化的，有：

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

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。

下面讨论的所有有关树上链邻域的问题均不包含广义点。

从圆到树上圆

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 $\mathrm{Path}(x, y)$ 都被称为点集 S 的直径。

直径中点和树上圆

定理

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）。

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 $K(S) = N(m(S), \frac{d(S)}{2})$。容易发现 $K(S)$ 必然是一个真实邻域。

点集、圆与点邻域的包含关系

定理

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 $\mathcal{K}(S) \subset N(u,r)$。

首先证明 $dis(u,m(S)) \le r - \frac{d(S)}{2}$。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 $dis(x,u) = dis(x,m(S)) + dis(m(S),u) = \frac{d(S)}{2} + dis(m(S),u) \le r$，因此 $dis(m(S),u) \le r - \frac{d(S)}{2}$。

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

树上邻域求并

考虑如下问题：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 $\mathcal{K}(S \cup T)$。

树上邻域求并

考虑如下问题：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 $dis(m(S), m(T)) + \frac{d(S)}{2} \le \frac{d(T)}{2}$，则显然 $\mathcal{K}(S \cup T) = \mathcal{K}(T)$。
- 若 $\mathcal{K}(T) \subset \mathcal{K}(S)$，即 $dis(m(S), m(T)) + \frac{d(T)}{2} \le \frac{d(S)}{2}$，则显然 $\mathcal{K}(S \cup T) = \mathcal{K}(S)$。

树上邻域求并

- 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 $\text{Path}(u_1, v_1)$，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 $\text{Path}(u_2, v_2)$，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。
中点为 $\frown(u_1, u_2, \frac{len+d(S)/2+d(T)/2}{2}) = \frown(m(S), m(T), \frac{len-d(S)/2+d(T)/2}{2})$。

例题

问题 (Range Diameter Sum)

给定一棵包含 n 个点的树，定义：

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

计算：

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

数据范围：$1 \le n \le 10^5$。

例题解法

求单个 $\operatorname{diam}(l,r)$ 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。

例题解法

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 $\mathcal{K}([mid+1, y])$。

例题解法

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：

- 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。

树上邻域求交

定理

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。

树上邻域求交

定理

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。

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：

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

树上邻域与链邻域的关系

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。

树上邻域与链邻域的关系

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。

定理

任意一个真实邻域都可以用链邻域来表示。

- 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。

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

对于树上任意两条链 $\mathrm{Path}(u_1, v_1), \mathrm{Path}(u_2, v_2)$，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 $\mathrm{Path}(u_1, v_1)$ 的子集，故必为若干段链。若段数 ≥ 2，则根据这些链也是 $\mathrm{Path}(u_2, v_2)$ 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）。 $\square$

等半径树上链邻域求交

对于树上的任意两条链 $\mathrm{Path}(u_1, v_1)$ 和 $\mathrm{Path}(u_2, v_2)$，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 (链相交情况)

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 $M(u_1, v_1, r) \cap M(u_2, v_2, r) = M(x, y, r)$。

引理 (链不交情况)

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。

等半径树上链邻域求交

定理 (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。

一般树上链邻域求交

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 (链相交情况)

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。

引理 (链不交情况)

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

一般树上链邻域求交

结合上面两个引理，可以得到：

定理 (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。

链邻域求交算法流程

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 $\curvearrowright (u, v, d)$。

链邻域求交算法流程

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 $d \leq dis(u, v)$，查询 $\curvearrowright (u, v, d)$。

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 $n = |V|$，容易简单做到 $O(n)$ 预处理，$O(\log n)$ 进行单次查询。

例题

问题 (Ald)

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：$1 \leq n, q \leq 10^5$。

例题解法

根据上面定理的结论，$\bigcap_{\mathrm{Path}(u,v)\in S} M(u,v,d)$ 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》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。

例题解法

根据链相交的等半径链邻域求交定理的结论，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 $\bigcap_{\text{Path}(u,v)\in S} \text{Path}(u,v) \neq \varnothing$，那么下式成立：

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

$$\text{where } \text{Path}(u',v') = \bigcap_{\text{Path}(u,v)\in S} \text{Path}(u,v)$$

例题解法

否则该结论并不成立。但是有如下结论成立:

引理

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:

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

例题解法

利用上述引理的结论，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

问题 (这里有只毛毛虫)

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：

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

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：

- 给定三个整数 u,v,c，将 $Cat(u,v,c)$ 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：$1\le n,q\le 5\times 10^5$。

例题解法

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 $\mathrm{Cat}(u, v, c)$，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。

例题解法

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交操作。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。

例题解法

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》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。

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，期待听众能够对部分理论进行进一步研究。

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。
- 感谢大家的聆听。

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.
3. 吴思成. (2023). 冬のエピローグ. Luogu.
4. 朱羿恺. (2023). Petrozavodsk Summer 2023. Day 7. PKU Contest Tutorials. QOJ.