浅谈信息学竞赛中的弦图问题

湖南省长沙市长郡中学 郭城志

摘 要

弦图是一类特殊的图，很多一般图上的 NPC 问题在弦图上都有优秀的解法。本文介绍了一些与弦图有关的知识，并通过几个例题展示了这些知识在信息学竞赛中的应用。

1 引言

弦图出现在信息学竞赛中已经有至少十年，但是没有得到广泛的普及，近几年出现在信息学竞赛中的弦图问题非常少，许多选手对弦图的了解也止步于基本定义和最大势算法的记忆上。本文较详细地介绍了一些与弦图有关的知识，希望能够帮助大家更加了解弦图。

本文第 2 节介绍了一些用到的记号和定义。

第 3 节介绍了一些弦图的基础知识。

第 4 节介绍了弦图的团树，以及团树、子树图、弦图之间的关系。

第 5 节通过几道例题，展示了弦图知识在信息学竞赛中的一些应用。

2 定义与约定

如无特殊说明，本文中的图均指无重边、自环的无向图。

对于图 $G = (V, E)$，令 $n = |V|$ 表示点集的大小，$m = |E|$ 表示边集的大小。为了方便，有时会直接将点从 $1 \dots n$ 编号。

定义 2.1 (邻域). 对于任意 $v \in V$，记 v 的邻域 $N(v) = \{u|(u, v) \in E\}$，即与 v 有边直接相连的点集。

定义 2.2 (导出子图). 对于任意 $A \subseteq V$，令 A 在 G 上的导出子图为 $G' = (A, E')$，其中 $E' = \{(u, v)|(u, v) \in E, u, v \in A\}$，即点集 A 和 G 中两端都在点集 A 中的边组成的子图。

定义 2.3 (团). 对于任意集合 $A \subseteq V$，A 是一个团当且仅当 $\forall u, v \in A, u \neq v$，有 $(u, v) \in E$，即点集 A 中任意两个不同的点都有边相连。如果不存在 $A' \supset A$ 使得 A' 是一个团，则称 A 为极大团。

定义 2.4 (弦图)**.** G 是弦图当且仅当对于 G 中任意一个长度大于 3 的简单环，都存在环上不相邻的两个点之间有边。环上不相邻的两个点之间的边也叫做弦。

3 基础知识

3.1 弦图的点割集

定义 3.1 (点割集)**.** 对于任意集合 $A \subseteq V$，定义 A 是 G 关于 u, v 两点的点割集，当且仅当 $u, v \notin A$，且 $V \setminus V'$ 在 G 上的导出子图中 u, v 不连通。如果不存在 $A' \subset A$ 满足 A' 也是 G 关于 u, v 的点割集，则称 A 是极小点割集。

关于弦图的极小点割集，有以下性质：

引理 3.1. 令 $G = (V, E)$ 是弦图，A 是 G 关于两个点 u, v 的极小点割集，则 A 是一个团。

证明. 令 $V \setminus A$ 中 u, v 所在的连通分量分别为 V_1, V_2。

显然，$\forall x \in A, N(x)$ 包含 V_1, V_2 中的点，否则删去 x 可以得到一个更小的点割集，与 A 是极小点割集矛盾。

那么，对于任意 $x, y \in A (x \neq y), N(x)$ 包含 V_1, V_2 中的点，$N(y)$ 也包含 V_1, V_2 中的点。令 x, y 之间在 V_1, V_2 内部的最短路径分别为 $x \to x_1 \rightsquigarrow y_1 \to y$ 和 $y \to y_2 \rightsquigarrow x_2 \to x$。那么，$x \to x_1 \rightsquigarrow y_1 \to y \to y_2 \rightsquigarrow x_2 \to x$ 是 G 中一个长度 > 3 的环。

因为 G 是弦图，所以该环上必然存在一条连接不相邻点的边。可以发现，如果该边不是 (x, y)，那么 $x \to x_1 \rightsquigarrow y_1 \to y$ 和 $y \to y_2 \rightsquigarrow x_2 \to x$ 两条路径中至少有一条不是最短路径，产生矛盾。因此边 (x, y) 存在，故结论成立。 □

3.2 弦图的单纯点

定义 3.2 (单纯点)**.** 对于图 $G = (V, E)$ 和任意 $v \in V$，v 是单纯点当且仅当 $N(v)$ 是一个团。

利用**引理 3.1**，我们可以证明一个结论：

引理 3.2. 所有弦图 $G = (V, E)$ 存在单纯点，不是完全图的弦图存在两个不相邻的单纯点。

证明. 按 n 归纳。$n = 1$ 时显然成立。

假设结论对于更小的 n 都成立。

G 是完全图或 G 不是连通图的情况是平凡的。

假设 G 不是完全图且 G 是连通图。任取两点 (u, v) 满足 $(u, v) \notin E$，令 A 为 G 关于 u, v 的极小点割集。根据**引理 3.1**，A 是一个团。令 $G \setminus A$ 中 u, v 所在的连通分量分别为 V_1, V_2。

令 $L = V_1 \cup A$。若 L 是一个团，那么 V_1 中任取一点都是 V_1 的导出子图的单纯点。否则，L 的导出子图中存在不相邻的单纯点 x, y，它们不可能都在 A 中，因为 A 是一个团。也就是说，x, y 至少有一个点在 V_1 中，它是 V_1 的导出子图的单纯点。

同理，V_2 的导出子图中也存在单纯点。又因为 A 是点割集，所以 V_1, V_2 之间不可能有边，我们就找到了 G 的两个不相邻的单纯点。 □

3.3 弦图的完美消除序列

定义 3.3 (完美消除序列)**.** *若 G 是弦图，则 G 的完美消除序列是一个点集的排列 $p_1, \ldots, p_n$，满足 $\forall i \in [1, n]$, $\{p_i\} \cup (N(p_i) \cap \{p_{i+1}, \ldots, p_n\})$ 是一个团。即对于排列中任意一个点，其自己和排列中在其后面的与其相邻的点是一个团。$\forall i \in [1, n]$，在给定完美消除序列的前提下，定义 $C(p_i) = \{p_i\} \cup (N(p_i) \cap \{p_{i+1}, \ldots, p_n\})$。*

每个弦图都存在完美消除序列。对于给定的弦图 $G = (V, E)$，可以使用最大势算法求出 G 的一个完美消除序列。该算法的流程为：对每个点 i 设置一个变量 $label_i$，初始值为 0。执行 n 轮以下过程：找到 i 使得 $label_i$ 最大，且点 i 还未被加入完美消除序列中，将点 i 加入到完美消除序列的最前面，然后 $\forall j \in N(i)$，将 $label_j$ 增加 1。

下面，我们证明该算法的正确性。不失一般性地，设最大势算法求出来的序列为 $1, 2, \ldots, n$。我们首先需要一个引理：

引理 3.3. *对于任意弦图 $G = (V, E)$，不存在元素两两不同的序列 $v_0, \ldots, v_k (k \geq 2)$ 满足：*

1. *$v_i, v_j \in E$ 当且仅当 $|i - j| = 1$。*
2. *$\forall i \in [1, k], v_0 > v_i$。*
3. *$v_1 < v_k$。*

证明. 假设存在这样的序列，那么有 $v_1 < v_k < v_0$，且 $(v_0, v_1) \in E, (v_0, v_k) \notin E$。后者说明，在最大势算法的过程中，$v_0$ 对 $label_{v_1}$ 有贡献，而对 $label_{v_k}$ 无贡献。为了使 v_k 比 v_1 先加入到完美消除序列，必存在 $x > v_k$，满足 $(x, v_k) \in E, (x, v_1) \notin E$。

任取一个这样的 x，并取最小的 $j \in [2, k]$ 满足 $(x, v_j) \in E$。那么 $(v_0, x) \notin E$，否则 $v_0 \to v_1 \to \cdots \to v_j \to x \to v_0$ 是一个长度 > 3 的无弦环。

如果 $x < v_0$，则 $v_0, \ldots, v_j, x$ 也是一个满足以上三条性质的序列，否则 $x, v_j, \ldots, v_0$ 也是满足性质的序列。每一种情况都使得序列最后一个元素增大，因此重复进行这个过程可以得到有无穷多个序列满足条件，产生矛盾。 □

假设存在 u, v, w 满足 $u < v < w, (u, v) \in E, (u, w) \in E, (v, w) \notin E$，那么 w, u, v 就是一个满足**引理 3.3** 中性质的序列，产生矛盾。因此最大势算法求出的是一个完美消除序列。

3.4 弦图的判定

有一个重要的弦图判定定理：

定理 3.1. 图 $G = (V, E)$ 是弦图的充要条件是，G 存在完美消除序列。

证明. 若 G 是弦图，我们可以通过最大势算法构造 G 的一个完美消除序列。

若 G 不是弦图，则 G 中存在一个长度 > 3 的环，满足环上不存在弦。假设完美消除序列存在，考虑环上在完美消除序列中最前面的点 v，其在环上与 v_1, v_2 直接相连。根据完美消除序列的定义，v_1, v_2 之间也应该有边直接相连，与环上不存在弦产生矛盾。 □

那么，我们可以利用完美消除序列的存在性来判定一个给定的图 $G = (V, E)$ 是否是弦图。对 G 运行最大势算法，如果 G 是弦图，那么我们可以求出 G 的一个完美消除序列；如果 G 不是弦图，我们求出来的就不是完美消除序列。

设最大势算法求出的序列为 $1, \ldots, n$，对于每个 i，设 $C(i) = \{i, p_1, \ldots, p_k\}$，其中 p 递增。若直接定义判定其是否是一个完美消除序列，我们需要判定 $C(i)$ 是否是一个团，这需要 $\Theta(k^2)$ 次判定，总复杂度 $O(nm)$。实际上，只需判定 p_1 是否与 $C(i)$ 中的其他点都有连边即可，因为假设我们是按 $n \ldots 1$ 的顺序依次枚举每个 i 的，那么根据归纳的思想，$p_j, p_k (j, k > 1)$ 是否有连边的判定，已经在 p_1 处完成了。这样，复杂度就降为了 $O(n + m)$。

3.5 弦图的极大团

令完美消除序列为 $1 \ldots n$，显然极大团必定是某个 $C(i)$。对于一个 i，$C(i)$ 不是极大团当且仅当存在 $j < i$ 满足 $C(i) \subset C(j)$。假设存在这样的 j，并存在 k 使得 $j < k < i$ 且 $(j, k) \in E$，那么 $C(i) \subset C(k)$ 同样成立。因此可以假设 $C(j) \setminus \{j\}$ 中最小的点为 i。

这种情况下，判定 $C(i) \subset C(j)$ 是否成立，显然只需判定 $|C(i)| + 1$ 是否等于 $C(j)$。因为 i 从 1 取到 n，$C(i) \setminus \{i\}$ 中最小的点总共只有 $O(n)$ 种，因此总复杂度为 $O(n + m)$。

3.6 弦图的色数/团数

定义 3.4 (色数)**.** G 的色数指的是最小的正整数 k，使得存在一种给 G 中每个点染上 $[1, k]$ 中的一种颜色的方案，满足每条边两端点颜色不同，记作 $\chi(G)$。

定义 3.5 (团数)**.** G 的最大团指的是大小最大的 $A \subseteq V$，满足 A 是一个团。G 的最大团的大小称为 G 的团数，记作 $\omega(G)$。

按照任意完美消除序列从后往前，给每个点染上未使用过的编号最小的颜色，使用的总颜色数 $k = \chi(G) = \omega(G)$。

证明. 显然 $\chi(G) \leq \omega(G)$。按照这种方法染色，显然 $k = \omega(G)$。而根据定义 $k \geq \chi(G)$，所以 $k = \chi(G) = \omega(G)$。 □

3.7 弦图的最大独立集/最小团覆盖

定义 3.6 (最大独立集)**.** G 的最大独立集指的是大小最大的 $A \subseteq V$，满足 A 中任意两点没有边相连。G 的最大独立集的大小记作 $\alpha(G)$。

定义 3.7 (最小团覆盖)**.** G 的最小团覆盖指的是用最少的团覆盖 G 中的所有点。使用的团的数量记作 $\kappa(G)$。

按照任意完美消除序列从前往后考虑每个点,若当前考虑的点与已被选入独立集中的点没有边相连,就将当前点加入独立集,最终得到的是最大独立集。设最大独立集为 $\{v_1, \ldots, v_k\}$，则最小团覆盖为 $\{C(v_1) \ldots, C(v_k)\}$。

证明. 首先，这些团确实是 G 的一个最小团覆盖。否则，若 G 中一个点不在该团覆盖内，说明完美消除序列中在其前面且与其有边的点都未被选择，那么其应该被加入独立集内。

其次，显然 $\alpha(G) \le \kappa(G), k \le \alpha(G), k \ge \kappa(G)$，因此 $k = \alpha(G) = \kappa(G)$。 □

4 子树图和团树

4.1 弦图的团树

定义 4.1 (团树)**.** 对于图 $G = (V, E)$，定义其团树为 $T = (\mu(G), E')$，满足对于任意 $v \in V$，$\mu_v(G)$ 在 T 上的导出子图连通。团树可能不存在。

每个弦图都存在团树。下面我们将给出一种方法，对于任意弦图 $G = (V, E)$，构造出其团树。

因为团树是一棵树，所以“团树上的团”这个描述没有意义。以下为了更加直观，我们把团树上的一个点，即 G 中的一个团也称为团树上的团。

求出 G 的任意一个完美消除序列，假设为 $1 \ldots n$。我们将按照完美消除序列从后往前增量构造，即对于 $i = n \ldots 1$，依次构造出 $\{i \ldots n\}$ 的导出子图的团树。在构造过程中，我们会令 G 中已加入的每个点 i 指向加入点 i 时团树上新产生的团。

假设我们已经对 $\{i + 1 \ldots n\}$ 的导出子图构造完毕，现在需要加入点 i。

令 j 为 $C(i) \setminus \{i\}$ 中最小的点。若不存在这样的 j，则将 $\{i\}$ 作为一个新团加入团树，与团树上任意一个团连边。

假设 j 存在。若 $|C(i)| = |C(j)| + 1$，并且 j 指向的团等于 $C(j)$，直接将 i 加入该团，即将团树上 $C(j)$ 替换成 $C(i)$。

否则，将 $C(i)$ 作为一个新团加入团树中，并与 j 指向的团连边。

容易发现算法流程中每个点 i 指向的团的大小是单调不降的，且加入点 i 时其指向的团是 $C(i)$。因此，判定 j 指向的团是否等于 $C(j)$，只需将其现在的大小与 $|C(j)|$ 比较即可。因此，该算法的时间复杂度为 $O(n + m)$。

下面，我们证明该算法的正确性：

考虑加入点 i 时三种不同的情况。

1. 点 i 是孤立点。这种情况下显然正确。

2. $|C(i)| = |C(j)| + 1$，并且 j 指向的团等于 $C(j)$。第一个条件说明 $C(i) = C(j) \cup \{i\}$，因为根据完美消除序列的定义，$C(i) \setminus \{i\} \subseteq C(j)$。第二个条件说明，$C(j)$ 是 $\{i+1, \ldots, n\}$ 的导出子图中的极大团。那么，$C(j)$ 不再是 $\{i, \ldots, n\}$ 的导出子图中的极大团，而 $C(i)$ 是新产生的一个极大团。而其他极大团在加入点 i 后仍然存在，因为点 i 的加入只可能使得是 $C(i) \setminus \{i\}$ 子集的极大团变成非极大团，而这些子集中除掉 $C(j)$ 以外的团在加入点 j 后必定已不是极大团。另一方面，得到的新树与原团树相比，区别仅仅在于某一个团内多出了新加入的点 i，因此包含每个点的团连通这一性质仍然满足。

3. 其他情况。按照类似的分析，所有 $\{i+1, \ldots, n\}$ 的导出子图中的极大团都不会变成非极大团；又因为 $C(i) \setminus \{i\} \subseteq C(j)$，包含每个点的团连通这一性质仍然满足。

4.2 子树图与弦图的关系

定义 4.2 (交图)**.** *考虑一个集合族 $\mathcal{F}$，其交图 $G = (V, E)$ 是这样一个图：每个点对应 $\mathcal{F}$ 中的一个集合，两个点之间有边当且仅当其对应的集合交集非空。*

定义 4.3 (子树)**.** *对于树 $T = (V, E)$，称 $V' \subseteq V$ 是 T 的子树，当且仅当 V' 在 T 上的导出子图是连通图。*

定义 4.4 (子树图)**.** *对于一棵树的子树族 $\mathcal{F}$，其交图称为子树图。*

引理 4.1. *若图 $G = (V, E)$ 是子树图，则 G 是弦图。*

证明. 令 G 是树 T 上子树族 $\mathcal{F}$ 的交图。我们将构造 G 的一个完美消除序列，以此证明 G 是弦图。

设 A_i 为点 i 对应的 $\mathcal{F}$ 中的子树。令 T 中任意一点为树根，将 V 中所有点排序，令排序后为 $1, \ldots, n$, 满足 $\forall i < j$, A_i 的根的深度不小于 A_j 的根的深度。因此，$\forall i < j$，若 $A_i \cap A_j \neq \varnothing$，则 A_j 必包含 A_i 的根。由此可得，$\forall i \in [1, n]$，满足 $j > i$ 且 $A_i \cap A_j \neq \varnothing$ 的 A_j 一定两两有交，故 $1, \ldots, n$ 是 G 的一个完美消除序列。

□

4.3 团树与子树图的关系

引理 4.2. *若图 $G = (V, E)$ 是存在团树，则 G 是子树图。*

证明. 令其团树为 T。令

$$\mathcal{F} = \{\mu_v(G) | v \in V\}$$

对于任意 $v \in V$，我们令 G 中的点 v 与 $\mathcal{F}$ 中的元素 $\mu_v(G)$ 相对应。我们将证明 G 是 $\mathcal{F}$ 的交图。

对于任意 $u, v \in V$，若 $\mu_u(G) \cap \mu_v(G) \neq \emptyset$，那么存在 G 中的极大团 A，满足 $A \in \mu_u(G) \cap \mu_v(G)$，即 $A \in \mu_u(G), A \in \mu_v(G)$。因此 $u, v \in A$，即 $(u, v) \in E$。

另一方面，如果 $(u, v) \in E$，那么存在极大团 A 满足 $u \in A, v \in A$（只需从 $\{u, v\}$ 开始任意扩展直到不能扩展为止，就可以得到一个这样的 A）。因此 $A \in \mu_u(G) \cap \mu_v(G)$，即 $\mu_u(G) \cap \mu_v(G) \neq \emptyset$。

那么，$(u, v) \in E$ 当且仅当 $\mu_u(G) \cap \mu_v(G) \neq \emptyset$，即 G 是 $\mathcal{F}$ 的交图。 □

最后，根据本节内容，我们可以得到一个定理：

定理 4.1. *对于图 $G = (V, E)$，以下三个命题等价：*

1. G 是子树图。

2. G 是弦图。

3. G 有团树。

5 应用

例题 1. *对于一个长度为 n 的字符串 s，定义其 $next$ 数组为一个长度为 n 的整数数组，其中 $next_i = \max\{j | j < i, s[1 \dots j] = s[i - j + 1 \dots i]\}$。*

现在给定一个长度为 n 的 $next$ 数组和字符集大小 k，求有多少字符串 s，满足 s 的 $next$ 数组为给定的 $next$ 数组。

$n \le 10^7$

模拟 KMP 算法的过程，我们会得到若干条限制，每条限制形如 s 的某两个字符必须相等，或 s 的某两个字符必须不相等。

考虑一个 n 个点的无向图 G，每个点对应 s 的一个字符。选择 s 每个位置填的字符就是给 G 中每个点染色；对于每一条限制，我们在两个字符对应的 G 中的点之间连形如两端点颜色必须相等/必须不相等的边。问题转化为：有多少种给 G 中每个点染上一种 $[1, k]$ 的颜色的方案，满足所有边的限制。

相等边是好处理的，将每一个由相等边连成的极大连通块缩成一个点即可。设缩点后得到的新图是 G'。现在的问题是，求 G' 有多少种染色方案，使得每条边两个端点的颜色不同。这是 NPC 问题，所以我们需要挖掘 G' 更多的性质。

分析 KMP 算法过程中求出的限制。考虑一棵树 T，对于每个 $i \in [2, n]$，T 中点 i 的父亲是 $next_{i-1} + 1$，点 1 是树根。限制实际上就是，某一些点 i 有一个祖先 f_i，i 的颜色必须和 f_i 的颜色相同，且和 i 到 f_i 路径上（不包括端点）的所有点颜色不同；另一些点与自己到根路径上（不包括自己）的所有点颜色不同。那么，一个相等边的极大连通块存在唯一一个块根，两个连通块有不等边当且仅当其块根有祖孙关系。

通过和**引理 4.1** 的证明类似的思想，容易发现，G' 是一个弦图，将所有相等块按照块根的编号从大到小排序就是一个完美消除序列。

怎样求弦图的 k 染色方案数呢？这是简单的：按完美消除序列从后往前依次决定每个点的颜色。假设当前考虑到了点 v，根据完美消除序列的定义，$C(v) \setminus \{v\}$ 是一个团，因此 $C(v) \setminus \{v\}$ 中的点一定被染成了两两不同的颜色。那么，点 v 可染的颜色数就是 $k - |C(v)| + 1$，答案就是每个点可染颜色数的乘积。

时间复杂度 $O(n)$。

例题 2. 给定弦图 $G = (V, E)$，*Alice* 和 *Bob* 将在 G 上博弈。

第一轮，*Alice* 选择不超过 k 个点，设选中的点集为 A_1，然后 *Bob* 选择 $V \setminus A_1$ 中的一个点，设其为 v_1。

第 $i(i > 1)$ 轮，*Alice* 选择不超过 k 个点，设选中的点集为 A_i，然后 *Bob* 选择 $V \setminus A_i$ 中的一个点，设其为 v_i。*Bob* 需要满足 v_{i-1} 和 v_i 之间存在一条不经过 $A_{i-1} \cap A_i$ 内的点的路径。

如果某一轮 *Bob* 无法选择 v_i，则 *Alice* 获胜。如果游戏能无限进行下去，则 *Bob* 获胜。

求出最小的 k，使得 *Alice* 能获胜。

$n, m \leq 10^5$

若 $k < \omega(G)$，显然 Bob 永远可以在最大团内选出一个点。

否则，考虑 G 的团树 T'，G 中的每一个点对应 T' 中的一个子树 $(\mu_v(G))$。原问题可以转化为一个树上的问题：给定树 T 和一个子树族 $\mathcal{F}$，每一轮 Alice 选择 $\mathcal{F}$ 中不超过 k 个子树 $\mathcal{A}_i$，然后 Bob 选择一个不在 $\mathcal{A}_i$ 中的子树 F_i，满足存在一个子树序列，以 F_{i-1} 开头，F_i 结尾，相邻子树有交，且不包含 $\mathcal{A}_{i-1} \cap \mathcal{A}_i$ 中的子树。并且，对于 T 中的每个点，$\mathcal{F}$ 中包含其的子树个数 $\leq k$。

第一步，Alice 可以选择所有包含 u 的子树，其中 u 是 T 中任意一个点。那么，删去点 u 后，T 将包含若干个连通分量，根据规则，Bob 必须选择一个完全在某个连通分量内的子树。假设 Bob 选择的子树所在的连通分量与 u 直接相连的点是 v。那么，在第二步，Alice 可以选择所有包含 v 的子树，删去 v 后，v 所在的连通分量又分成若干个更小的连通分量。容易发现，Bob 再次选择的子树只能完全在这些更小的连通分量中……以此类推，最终 Bob 一定会无法选择子树。

所以，答案就是 $\omega(G)$。时间复杂度 $O(n + m)$。

例题 3. 给定弦图 $G = (V, E)$，求至少删去多少个点，才能使得 G 中不存在环。

$n, m \leq 10^5$

弦图删去若干个点后还是弦图，所以不存在环等价于每个极大团的大小都 ≤ 2。

考虑求最多能保留的点数。建出 G 的团树 T'，限制就是 T' 中的每个团内至多只能保留两个 G 中的点。

接下来，我们可以将原问题作和**例题 2** 一样的转化，不过此处额外用到了一个团树的性质：每个点代表原图的一个极大团。问题转化为：给定树 T 和一个子树族 $\mathcal{F}$，需要从 $\mathcal{F}$ 中选出尽可能多的子树，满足对于 T 上的每个点 u，选出的包含 u 的子树个数不超过 2。

转化后的问题可以用一个简单的树形 DP 解决。将 T 以任意点为根，设 S_u 表示 T 中所有到根的简单路径经过点 u 的点，设 $f_{u,i,j}$ 表示考虑完了 $\mathcal{F}$ 中所有与 S_u 有交的子树，选出的包含 u 的子树分别为 i, j（其中 i, j 可以为 0，表示选出的包含 u 的子树个数小于 2），最多能选出的子树个数。DP 转移需要满足：对于一个 u 的儿子 v，若 $i \neq 0$ 且子树 i 包含点 v，则从 $f_{v,i',j'}$ 转移需要保证 $i' = i$ 或 $j' = i$；同样地，若 $j \neq 0$ 且子树 j 包含点 v，则需要保证 $i' = j$ 或 $j' = j$。

根据团树的构建过程，可以得到 T' 上每个团的大小之和是 $O(m)$ 的，并且每个团的大小是 $O(\sqrt{m})$ 的（因为若存在一个大小为 t 的团，则团内有 $\Theta(t^2)$ 条边，而总边数只有 m）。也就是说，$\mathcal{F}$ 中每个子树的大小之和是 $O(m)$ 的，且对于 T 中每个点 u，$\mathcal{F}$ 中包含 u 的子树个数是 $O(\sqrt{m})$ 的。因此，DP 的总状态数是 $O(m\sqrt{m})$ 的，若精细实现使得每一次转移 $O(1)$ 完成，总时间复杂度即为 $O(m\sqrt{m})$。

6 总结

本文第 3 节讲述了一些弦图的基础知识，体现出了弦图有很多优秀的性质，一些在一般图上难以解决的问题在弦图上可以轻易解决。

第 4 节介绍的团树给出了一种化弦图为树的方法，利用团树我们可以更好地分析弦图的结构，解决更多的问题；同时也说明了只有弦图存在团树，该方式难以被扩展到一般图上解决问题。

第 5 节的几个例题介绍了第 3, 4 节知识的一些小应用。

事实上，因为作者的水平有限，这篇文章介绍的内容还相当浅，弦图还有很多可以被挖掘的东西。希望本文能够起到一个抛砖引玉的作用，吸引更多的读者来研究弦图，让更多优秀的弦图题出现在信息学竞赛中。

7 致谢

感谢中国计算机学会提供学习和交流的平台。

感谢国家集训队教练高闻远的指导。

感谢长郡中学谢秋锋老师的关心和指导。

感谢彭思进同学、高子翼同学对我的启发和对本文的帮助。

感谢父母对我的关心与支持。

参考文献

[1] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. 3. Combin. Theory Ser. 8, 16:47-5G, 1974.

[2] Spinrad, J.P. Efficient Graph Representations; Fields Institute Monographs, American Mathematical Society: Presidence, RI, USA, 2003.

[3] Wikipedia,Chordal_graph

[4] OI Wiki, 弦图