國 立 交 通 大 學

應用數學系
碩 士 論 文

超立方體及局部扭轉超立方體之獨立擴張樹之建造

Constructing independent spanning trees for hypercubes and locally twisted cubes

研 究 生：劉宜君
指導教授：陳秋媛 教授

中 華 民 國 九 十 八 年 六 月

致　　謝

求學過程中，每進一個新的階段都會到不同縣市就讀，新竹市是我唯一例外的縣市！倒也不是因為新竹市有多迷人，而是國立交通大學應用數學系真的是個相當棒的學習環境。系上的教授個個學識淵博、關心學生，而且同學之間都會相互關心照應。兩年，如白駒過隙般的短暫！還記得大家一起修課、一起在教室討論著作業該如何寫、問題該如何解。當我們想破了一顆又一顆的腦袋後還是想不透時，大家就默默的跑去找教授討論的情景。在這樣常常一起討論的日子裡，大家一個一個上台發表自己的想法與解法的同時、累了休息打屁聊天的同時，一起唸書準備考試的同時，我感覺到了同學之間牢不可破的情誼與學習的快樂。感謝組合組所有的同學讓我覺得修課是如此的開心、與同學的聯繫是如此的緊密。還要感謝系上所有的教授細心的指導，尤其是傅恆霖教授總是能簡單明瞭的傳授所有的知識！還有符麥克教授，我修課期間常常去研究室打擾到很晚，但教授還是很開心且不厭其煩的講解與指導，真的非常的感恩。也有幾次不小心模仿了教授被發現但卻笑笑的就過了，所以在此也要感謝教授的寬宏大量呢！

再來要感謝我的指導教授陳秋媛以及同門的學長、同學與學弟們。秋媛教授非常關心我們的生活與態度，也很耐心的指導我們研究！以前學習課內知識對我來說就只是容易與不容易的分別而已。然而在教授的指導之下學習與研究居然變的有趣，讓我願意主動去想更多而不只是單純接收知識而已。有些原本對我來說不容易且也不喜歡的部份，也讓我不害怕進而願意主動去學習，而不只是被動消極的認為自己的能力就只能如此。以上的經驗與成長對我來說都是相當重要的！還要感謝國元學長不吝於討論與幫忙，看到覺得不錯且適合我的論文還會印給我；感謝鈺傑學長常常跑來研究室關心我們的研究，還會幫忙看或提供想法！感謝志文、子宏、信昌、松育、慧棻、士慶、思綸、思賢與健峰，研究路上的討論與相伴。還有其他的研究生們，舜婷、帥氣簡科科文昱大哥、小張、阿華田、娃娃音、光祥、哲皓、冠霖與廷彰等，平常都受到你們的照顧了。

還要感謝諮商中心的志工們！在學期間常會跑到中心的沙發打滾、哀嚎還有休息，中心的大家總是能給予關心與支持，讓我又充滿了向前的力氣。尤其是怡文、小翊常陪我聊天，還有丙丙跟我組成的研究生支持團體。你們是我在這段最忙亂、最低潮的時間裡最重要的支柱，當然也還要感謝三不五時就會被我吵的輔國與敬達還有我那去當兵的好姐妹昱丞，你們都是很重要的朋友。當然不能忘了常常會為大家帶來三餐的柏佑啦。

也要感謝一直都關心與掛念著我的國高中朋友們，謝死豬、玉米、小狐貍、俞秀、秀貞、淑媛、靜蘭還有懷升；感謝大學好友中偶而想到會來找我的笨大樹還有會帶食物給我的小賢。還有佔了很重~份量的、常常餵胖我，但少少帶我出去玩樂運動的小馬，雖然當兵變笨後偶而會覺得有點煩，但平常都很照顧、關心與包容我。

最後要感謝我父母的支持讓我無後顧之憂的念書，也把我養的頭好壯壯使我擁有耐操健壯的身心面對未來的路。還有常會稍來問候的姐姐、三不五時就很愛找我聊天的妹妹還有悶騷愛裝酷的帥弟，你們是我最大的後盾。雖然當我們發現論文要改的東西實在太多了卻不能改天，不過當要感謝的人太多了，總是可以謝天的吧！感謝上天賜給我這麼多關心我的人還有這麼好的環境讓我成長學習。在此謹以寥寥數句表達感謝之意！

超立方體及局部扭轉超立方體之獨立擴張樹之建造

研究生：劉宜君　　　　　　　　　　指導老師：陳秋媛　教授

國 立 交 通 大 學

應 用 數 學 系

摘　要

在網路中使用多棵獨立擴張樹對於資料廣播有相當多的好處，例如：可以提高容錯以及頻寬等；因此，在各種的網路結構上，建造多棵獨立擴張樹，一直以來都被廣泛地研究。Zehavi 和 Itai 在文獻[26]中，對於建造多棵獨立擴張樹提出了兩個猜測。「點猜測」闡述的是：在一個點連通度為 n 的圖上，能以圖中任一點為樹根，產生 n 棵點獨立擴張樹；「邊猜測」闡述的是：在一個邊連通度為 n 的圖上，能以圖中任一點為樹根，產生 n 棵邊獨立擴張樹。在文獻[16] 中，Khuller 和 Schieber 證明了點猜測能涵蓋邊猜測。局部扭轉超立方體是超立方體的變形。最近，Hsieh 和 Tu 在文獻[10]中，提出了一個能在 n 維局部扭轉超立方體上，建造以 0 為樹根的 n 棵邊獨立擴張樹的演算法。因為局部扭轉超立方體不具點對稱性質，Hsieh 和 Tu 所提出的演算法無法解決局部扭轉超立方體的邊猜測。在這篇論文中，我們提出了一個可以在局部扭轉超立方體上，以任一點為樹根，建構 n 棵點獨立擴張樹的演算法；我們的演算法證明了局部扭轉超立方體符合點猜測，當然，也證明了局部扭轉超立方體符合邊猜測。此外，我們的演算法也能在超立方體上得到一樣的結果。

關鍵詞：資料廣播、演算法設計與分析、點獨立擴張樹、局部扭轉超立方體、超立方體、平行演算法。

中 華 民 國 九 十 八 年 六 月

Constructing independent spanning trees for hypercubes and locally twisted cubes

Student: Yi-Jiun Liu Advisor: Chiuyuan Chen

Department of Applied Mathematics
National Chiao Tung University
Hsinchu, Taiwan 30050

Abstract

The use of multiple independent spanning trees (ISTs) for data broadcasting in networks provides a number of advantages such as the increase of fault-tolerance and bandwidth. Thus the designs of multiple ISTs in several classes of networks have been widely investigated. In [27], Zehavi and Itai stated two versions of the n independent spanning trees conjecture. The vertex (edge) conjecture is that any n-connected (n-edge-connected) graph has n vertex-ISTs (edge-ISTs) rooted at an arbitrary vertex r. In [16], Khuller and Schieber proved that the vertex conjecture implies the edge conjecture. Recently, in [12], Hsieh and Tu proposed an algorithm to construct n edge-ISTs rooted at vertex 0 for an n-dimensional locally twisted cube LTQ_n, which is a variant of the hypercube. Since LTQ_n is it not vertex-transitive, Hsieh and Tu's result does not solve the edge conjecture for the locally twisted cube. In the thesis, we confirm the vertex conjecture (and hence also the edge conjecture) for the locally twisted cube by proposing an algorithm to construct n vertex-ISTs rooted at any vertex for the LTQ_n. We also confirm the vertex conjecture (and hence also the edge conjecture) for the hypercube.

Keywords: Data broadcasting; Design and analysis of algorithms; Vertex-disjoint spanning trees; Locally twisted cubes; Hypercubes; Parallel algorithm.

Contents

List of Figures

List of Tables

1 Introduction

This thesis considers the problem of constructing n independent spanning trees rooted at an arbitrary vertex of an n-dimensional locally twisted cube or hypercube. Graph terminology and notation used in this thesis are standard; see [4] and [23] except as indicated.

All graphs in this thesis are simple undirected graphs. Let G be a graph with vertex set $V(G)$ and the edge set $E(G)$. Let $x, y \in V(G)$. A path from x to y is denoted as x, y-path. Two x, y-paths P and Q are *internally edge-disjoint* if $E(P) \cap E(Q) = \emptyset$. Two x, y-paths P and Q are *internally vertex-disjoint* if they are internally edge-disjoint and $V(P) \cap V(Q) = \{x, y\}$. A subgraph T of G is a *spanning tree* of G if T is a tree and $V(T) = V(G)$. Two spanning trees T and T' of G are *vertex-independent* (resp., *edge-independent*) if T and T' are rooted at the same vertex, say r, and for each $v \in V(G)$, the r, v-path in T and the r, v-path in T' are internally vertex-disjoint (resp., internally edge-disjoint). A set of spanning trees of G are *vertex-independent* (resp., *edge-independent*) if they are pairwise vertex-independent (resp., pairwise edge-independent).

Recently, the problems of constructing multiple vertex-independent spanning trees (vertex-ISTs) and constructing multiple edge-independent spanning trees (edge-ISTs) for a given graph have received much attention. In [27], Zehavi and Itai stated two versions of the n independent spanning trees conjecture:

(Vertex Conjecture) Any n-connected graph has n vertex-ISTs rooted at an arbitrary vertex r.

(Edge Conjecture) Any n-edge-connected graph has n edge-ISTs rooted at an arbitrary vertex r.

Zehavi and Itai [27] also raised the question: It would be interesting to show that either the vertex conjecture implies the edge conjecture, or vice versa. Later, Khuller and Schieber [16] successfully proved that the vertex conjecture implies the edge conjecture, i.e., if any n-connected graph has n vertex-ISTs, then any n-edge-connected graph has n

edge-ISTs. Khuller and Schieber's proof also works for the directed case. For the directed case, Edmonds [7] solved the edge conjecture. Khuller and Schieber [16] pointed out that the vertex conjecture for directed graphs is the strongest conjecture since it implies all the other conjectures.

The vertex and the edge conjectures have been confirmed only for $n \leq 4$. In [15], Itai and Rodeh gave a linear-time algorithm for constructing two edge-ISTs in a 2-edge-connected graph; they also solved the vertex conjecture for $n = 2$. In [27], Zehavi and Itai solved the vertex conjecture for $n = 3$, but they did not proposed an algorithm for constructing three vertex-ISTs. In [6], Cheriyan and Maheshwari proposed an $O(|V(G)|^2)$-time algorithm for constructing three vertex-ISTs in a 3-connected graph. In [5], Curran et al. proposed an $O(|V(G)|^3)$-time algorithm for constructing four vertex-ISTs in a 4-connected graph. When $n \geq 5$, both the vertex and the edge conjectures are still open. It has been proven that the vertex (or the edge) conjecture holds for several restricted classes of graphs or digraphs, such as planar graphs [9, 10, 17, 18], maximal planar graphs [19], product graphs [20], chordal rings [14, 24], de Bruijn and Kautz digraphs [8, 11], and hypercubes [22, 26].

The design of vertex- and edge-ISTs has applications to reliable communication protocols. For example, a rooted spanning tree in the underlying graph of a network can be viewed as a broadcasting scheme for data communication and fault-tolerance can be achieved by sending n copies of the message along the n independent spanning trees rooted at the source node [1]. For other applications, see [3] for the multi-node broadcasting problem, [21] for one-to-all broadcasting, and [2] for n-channel graphs, reliable broadcasting, and secure message distribution.

This thesis considers the problem of constructing n vertex-ISTs rooted at an arbitrary vertex of an n-dimensional locally twisted cube LTQ_n or an n-dimensional hypercube Q_n (these cubes will be defined later). Since we focus on vertex-ISTs, in the remaining discussion, we will simply use ISTs to denote vertex-ISTs unless otherwise specified. Note that the development of algorithms for constructing ISTs tends toward pursuing two research goals: one is to design efficient construction schemes (for example, [14, 17, 19, 24] propose linear-time algorithms) and the other is to reduce the heights of ISTs (for example, [11, 22, 24] propose the idea of height improvements). Let G be an n-connected graph, let T be a spanning tree of G rooted at vertex r, and let $d(T; r, v)$ denote the depth of vertex v in T. The *average path length* of a set $\mathcal{S} = \{T_0, T_1, \dots, T_{n-1}\}$ of n ISTs rooted at vertex r in G is defined to be

$$\textstyle\sum_{i=0}^{n-1} \sum_{v \in V(G)\setminus\{r\}} d(T_i; r, v)/n.$$

A set $\mathcal{S}$ of n ISTs rooted at vertex r in G (if this set exists) is called *optimal* if the average path length of $\mathcal{S}$ is the minimum among all possible sets of n ISTs rooted at r in G.

The hypercube is one of the most popular interconnection networks due to its simple structure and ease of implementation. However, it has been shown that the hypercube does not achieve the smallest possible diameter for its resources. Therefore, many variants of the hypercube have been proposed. The most well-known variants are twisted cubes (TQ), crossed cubes (CQ), and Möbius cubes (MQ), and locally twisted cubes (LTQ). In the following table, we list the connectivity, edge-connectivity and diameters of Q_n and its variants. It is well known that a hypercube Q_n is n-connected. Since Q_n is itself a

Table 1: The connectivity, edge-connectivity and diameters of Q_n and its variants.

$G\backslash$properties	$\kappa(G)$	$\lambda(G)$	diameter
Q_n	n	n	n
TQ_n	n	n	$\lceil (n+1)/2 \rceil$
CQ_n	n	n	$\lceil (n+1)/2 \rceil$
MQ_n	n	n	in 0-MQ_n, $\lceil (n+2)/2 \rceil$ for $n \geq 4$ in 1-MQ_n, $\lceil (n+1)/2 \rceil$ for $n \geq 1$
LTQ_n	n	n	2 if $n = 3$ 3 if $n = 4$ $\lceil (n+3)/2 \rceil$ if $n \geq 5$

product graph, the algorithm proposed by Obokata et al. [20] can be used to construct n ISTs for Q_n. As to the construction of the height-reduced ISTs on Q_n, Tang et al. [22] modified the algorithm in [20] and proposed an $O(n2^n)$-time algorithm for constructing an optimal set of n ISTs for hypercubes Q_n. It was pointed out by Yang et al. [26] that the algorithms in [20] and [22] are designed by a recursive fashion and such a construction forbids the possibility that the algorithm could be parallelized; Yang et al. therefore proposed a parallel construction for an optimal set of n ISTs for Q_n.

Although Q_n is a product graph, it is not known whether its variants are also product graphs. For example, it is not known whether the locally twisted cube LTQ_n is a product graph. The locally twisted cube was proposed by Yang et al. in [25]; the motivation of proposing such a variant is that a better hypercube variant should be conceptually closer to hypercube than other existing variants. In locally twisted cubes, the labels of any two adjacent vertices differ in at most two successive bits. In [12], Hsieh and Tu proposed an algorithm to construct n edge-ISTs for LTQ_n. Do notice that Hsieh and Tu did not solve the edge conjecture for the locally twisted cube since their algorithm uses vertex 0 as the common root of edge-ISTs and a locally twisted cube is *not* vertex-transitive. For example, in LTQ_5, vertex 1 can reach any vertex within 3 steps but vertex 0 has to take 4 steps to reach vertex 30.

The sequential algorithm in [22] and the parallel algorithm in [26] obtain an optimal set of n ISTs. However, these algorithms work only for hypercubes. In this thesis, we outline an approach to construct n vertex-ISTs rooted at an arbitrary vertex of an n-dimensional locally twisted cube or hypercube. Thus we confirm both the vertex and the edge conjectures for the locally twisted cube and hypercube.

This thesis is organized as follows. In Section 2, we give some definitions and notations. In Section 3, we outline an approach to construct n vertex-ISTs rooted at an arbitrary vertex of an n-dimensional general cube. In Sections 4 and 5, we prove that our approach constructs n ISTs for LTQ_n and Q_n, respectively. The final section concludes this thesis.

2 Some preliminaries and our algorithm

In the remaining discussion, $\oplus$ denotes the bitwise XOR operation. As a reference,

$$0 \oplus 0 = 0,\ 0 \oplus 1 = 1,\ 1 \oplus 0 = 1,\ 1 \oplus 1 = 0.$$

If $u = (u_{n-1}\ u_{n-2}\ \cdots\ u_0)_2$ and $v = (v_{n-1}\ v_{n-2}\ \cdots\ v_0)_2$, then we define

$$u \oplus v = (u_{n-1} \oplus v_{n-1} \quad u_{n-2} \oplus v_{n-2} \quad \cdots \quad u_0 \oplus v_0)_2.$$

Also, $u \oplus v \oplus w = (u \oplus v) \oplus w$.

The *n-dimensional hypercube* Q_n is a graph with 2^n vertices and $n \cdot 2^{n-1}$ edges such that its vertices are n-tuples with entries in $\{0, 1\}$ and its edges are the pairs of n-tuples that differ in exactly one position. Thus Q_1 is the complete graph with two vertices 0 and 1, and Q_n $(n \geq 2)$ is built from two copies of Q_{n-1} as follows: Let $k \in \{0, 1\}$ and let kQ_{n-1} denote the graph obtained by prefixing the label of each vertex of one copy of Q_{n-1} with k; connect each vertex $(0x_{n-2} \ldots x_1x_0)_2$ of $0Q_{n-1}$ with the vertex $(1x_{n-2} \ldots x_1x_0)_2$ of $1Q_{n-1}$ by an edge.

We now define a generalization of Q_n. The *n-dimensional general cube* GQ_n is defined recursively as follows (see Figure 1). GQ_1 is Q_1, and GQ_n $(n \geq 2)$ is built from two GQ_{n-1}'s (not necessarily identical) as follows: Let $k \in \{0, 1\}$ and let kGQ_{n-1} denote the graph obtained by prefixing the label of each vertex of one of the two GQ_{n-1}'s with k; add a perfect matching between $0GQ_{n-1}$ and $1GQ_{n-1}$, i.e., each vertex in $0GQ_{n-1}$ is adjacent to exactly one vertex in $1GQ_{n-1}$.

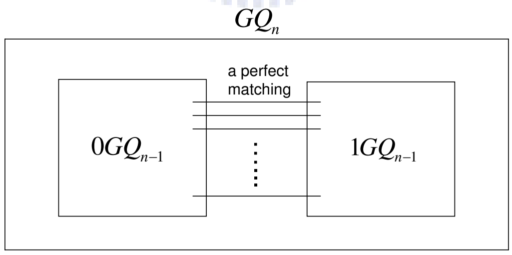

Figure 1: The n-dimensional general cube GQ_n.

The *n-dimensional locally twisted cube* LTQ_n is defined recursively as follow. LTQ_1 is Q_1, and LTQ_2 is the graph consisting of four vertices labeled with 00, 01, 10, and 11, respectively, and connected by the four edges (00, 01) (00, 10), (01, 11), and (10, 11). LTQ_n $(n \geq 3)$ is built from two identical LTQ_{n-1}'s as follows: connect each vertex $(0x_{n-2}x_{n-3} \cdots x_0)_2$ of $0LTQ_{n-1}$ with the vertex $(1(x_{n-2} \oplus x_0)x_{n-3} \cdots x_0)_2$ of $1LTQ_{n-1}$ by an edge. See Figures 2 and 3.

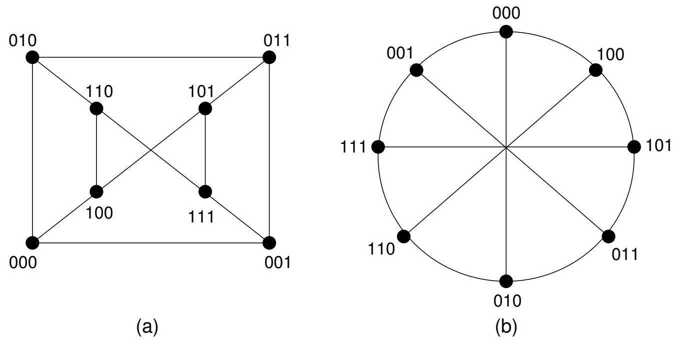

Figure 2: (a) LTQ_3. (b) A symmetric drawing of LTQ_3.

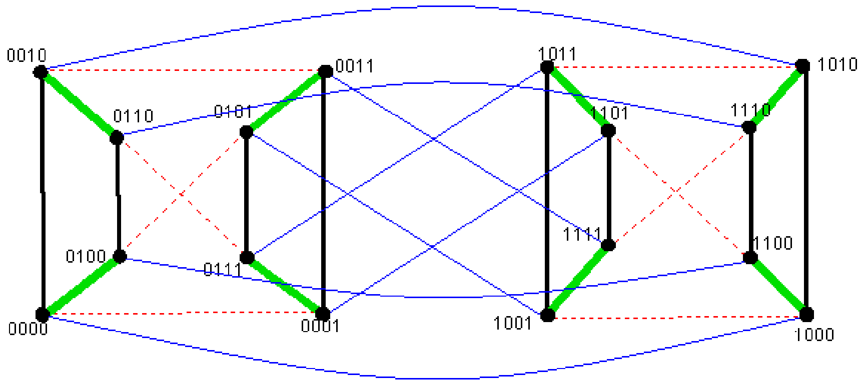

Figure 3: LTQ_4 and its $\mathcal{F} = \{f_3, f_2, f_1, f_0\}$. The ordinary lines (depicted as color blue), the most solid lines (depicted as color green), the second solid lines (depicted as color black), and the dashed lines (depicted as color red) are perfect matchings f_3, f_2, f_1, and f_0, respectively.

We assume conventionality of the vertex prefixing method kGQ_{n-1} which will be used repeatedly in the definitions of specific hypercube variants late in this thesis unless otherwise specified. It have been shown that crossed cubes, Möbius cubes, and locally twisted cubes are the examples of GQ_n; see [13]. Note that the two GQ_{n-1}'s in GQ_n are not necessarily identical. For instance, for crossed cubes and locally twisted cubes, the two GQ_{n-1}'s are identical; but for Möbius cubes, they are not.

Recall that GQ_n ($n \geq 2$) is built recursively by adding a perfect matching between $0GQ_{n-1}$ and $1GQ_{n-1}$; denote this perfect matching by f_{n-1}. Then $0GQ_{n-1}$ and $1GQ_{n-1}$ are built recursively by adding a perfect matching between $00GQ_{n-2} \cup 10GQ_{n-2}$ and $01GQ_{n-2} \cup 11GQ_{n-2}$; denote this perfect matching by f_{n-2}. Here $\cup$ means the *union* of graphs. Specifically, $G \cup H$ is the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$. In general, let us define $L^i = \{b_1 b_2 \cdots b_i | b_j \in \{0,1\}$ for $j = 1, 2, \ldots, i\}$ and $L^0 = \epsilon$ (the empty string). Let f_{n-i} denote the perfect matching between $\bigcup_{x \in L^{i-1}} x0GQ_{n-i}$ and $\bigcup_{x \in L^{i-1}} x1GQ_{n-i}$ and let

$$\mathcal{F} = \{f_{n-1}, f_{n-2}, \ldots, f_0\}$$

be the set of perfect matchings used to build GQ_n. See Figure 3 for an illustration of $\mathcal{F}$.

The following lemma is obvious and its proof is omitted.

Lemma 1. *Let $\mathcal{F} = \{f_{n-1}, f_{n-2}, \ldots, f_0\}$ be the set of perfect matchings used to build GQ_n. Then for every $0 \leq i < n$ and every $v = (v_{n-1}v_{n-2} \cdots v_0)_2 \in V(GQ_n)$, f_i will not affect bits $v_{n-1}, v_{n-2}, \ldots, v_{i+1}$ of v.*

Before going further, we give the $\mathcal{F}$ of the hypercube and the locally twisted cube. For convenience, $\overline{v}_i$ denotes the complement of v_i. First consider Q_n. Let $v = (v_{n-1}v_{n-2} \cdots v_0)_2 \in V(Q_n)$. Then $\mathcal{F} = \{f_{n-1}, f_{n-2}, \ldots, f_0\}$ in which f_i is defined by

$$f_i(v) = (v_{n-1}v_{n-2} \cdots v_{i+1}\overline{v}_i v_{i-1} \cdots v_0)_2. \tag{1}$$

Now consider the locally twisted cube. The adjacency relation of LTQ_n has been worked out by [13]; see the following.

Lemma 2. [13] *For every $v = (v_{n-1}v_{n-2} \ldots v_0)_2 \in V(LTQ_n)$, the n vertices $y_0, y_1, \ldots, y_{n-1}$ adjacent to v are:*

$$\begin{aligned}
&y_0 = (v_{n-1}v_{n-2}v_{n-3} \cdots v_2 v_1 \overline{v}_0)_2,\\
&y_1 = (v_{n-1}v_{n-2}v_{n-3} \cdots v_2 \overline{v}_1 v_0)_2,\\
&y_2 = (v_{n-1}v_{n-2}v_{n-3} \cdots \overline{v}_2\ (v_1 \oplus v_0)\ v_0)_2,\\
&\vdots\\
&y_{n-2} = (v_{n-1}\overline{v}_{n-2}\ (v_{n-3} \oplus v_0) v_{n-4}\ \cdots v_1 v_0)_2,\\
&y_{n-1} = (\overline{v}_{n-1}\ (v_{n-2} \oplus v_0)\ v_{n-3} \cdots v_2 v_1 v_0)_2.
\end{aligned}$$

Let $v = (v_{n-1}v_{n-2} \cdots v_0)_2 \in V(LTQ_n)$. By Lemma 2, $\mathcal{F} = \{f_{n-1}, f_{n-2}, \ldots, f_0\}$ of LTQ_n is defined by

$$f_i(v) = \begin{cases} (v_{n-1}v_{n-2} \cdots v_1 \overline{v}_0)_2 & \text{if } i = 0,\\ (v_{n-1}v_{n-2} \cdots v_2 \overline{v}_1 v_0)_2 & \text{if } i = 1,\\ (v_{n-1}v_{n-2} \cdots v_{i+1}\overline{v}_i\ (v_{i-1} \oplus v_0)\ v_{i-2}v_{i-3} \cdots v_0)_2 & \text{if } 2 \leq i \leq n-1. \end{cases} \tag{2}$$

In the remaining discussion, (u, v) denotes the edge between u and v; $T_0, T_1, \ldots, T_{n-1}$ denote subsets of edges of the given GQ_n; and r denotes the root of n IST. The vertex $f_i(r)$ will be the son of r in T_i. Moreover, $v \in V(T_i)$ means that v is an endpoint of an edge in T_i, and $v \in V(T_i) \setminus \{r, f_i(r)\}$ means that $v \in V(T_i)$ and v is neither the root nor the son of the root. Now we are ready to propose an algorithm for constructing n ISTs of a given GQ_n.

Algorithm 1 Construct n ISTs for GQ_n.

Input: $\mathcal{F} = \{f_{n-1}, f_{n-2}, \ldots, f_0\}$ used to build the given GQ_n and a vertex r of the GQ_n.
Output: n ISTs $T_0, T_1, \ldots, T_{n-1}$ rooted at r.

1: **for** each processor i $(0 \leq i \leq n)$ **do in parallel**
2: $\quad son \leftarrow f_i(r)$;
3: $\quad S \leftarrow \{son\}$;
4: $\quad$ **for** $m = i + 1$ to $i + n$ **do**
5: $\quad\quad S' \leftarrow \emptyset$;
6: $\quad\quad$ **for** each vertex $v \in S$ **do**
7: $\quad\quad\quad u \leftarrow f_{m \bmod n}(v)$;
8: $\quad\quad\quad T_i \leftarrow T_i \cup \{(v, u)\}$;
9: $\quad\quad\quad S' = S' \cup \{u\}$;
10: $\quad\quad$ **endfor**
11: $\quad\quad S \leftarrow S \cup S'$;
12: $\quad$ **endfor**
13: **end for**

Call the for-loop in lines 4 to 12 in the algorithm the outer for-loop for convenience. Also, call the for-loop in lines 6 to 10 in the algorithm the inner for-loop for convenience. Two examples of Algorithm 1 are given in Figure 4. If we replace **do in parallel** with **do in sequential**, then Algorithm 1 becomes a sequential algorithm. If a top-down fashion is insisted on, then Algorithm 1 can be modified to Algorithm 2 by adding lines 3, 14∼16 and replacing $i + n$ with $i + n - 1$ in Algorithm 1. Algorithm 2 builds n ISTs of a GQ_n in a top-down fashion; the algorithms in [12, 26] construct spanning trees in a bottom-up fashion. A top-down fashion is preferred since these n ISTs are used for broadcasting messages from the top (the root) of the trees.

We have a lemma.

Lemma 3. *For each $i \in \{0, 1, \ldots, n-1\}$, T_i constructed by Algorithm 1 has the properties that*

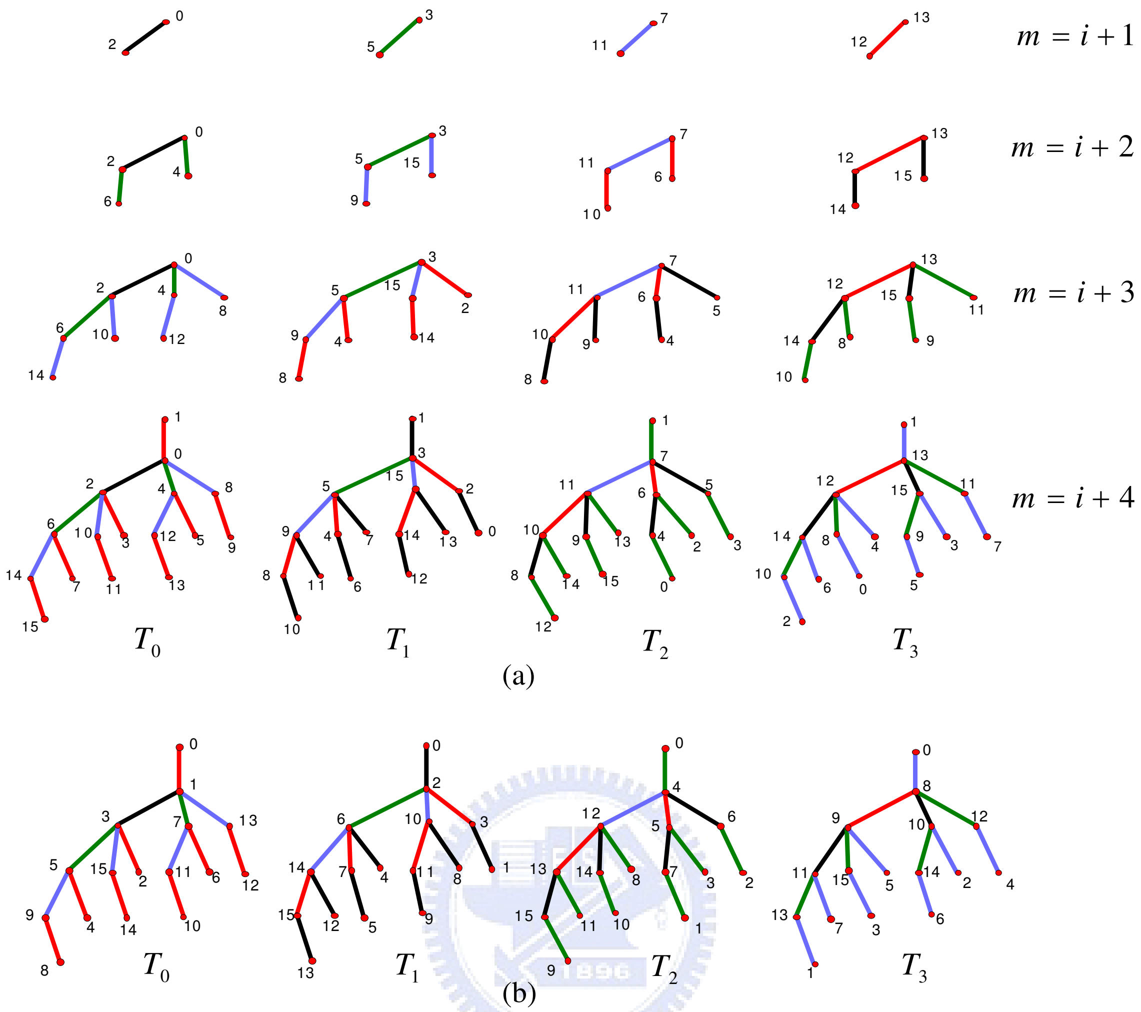

Figure 4: Two examples of Algorithm 1: constructing 4 ISTs T_0, T_1, T_2 and T_3 for LTQ_4. The edges depicted as color red are obtained from f_0, color black are from f_1, color green are from f_2, and color blue are from f_3. (a) The common root is 1. (b) The common root is 0.

(i) $(r, f_i(r)) \in T_i$;

(ii) *for each* $v \in V(G) \setminus \{r, f_i(r)\}$, *if* $v \in V(T_i)$, *then the path from* $f_i(r)$ *to* v *in* T_i *uses each perfect matching in* $\mathcal{F}$ *at most once.*

Proof. Property (i) follows from line 3. Property (ii) follows from the fact that $f_{m \bmod n}$ used in the for-loop between lines 7 and 11 are distinct. ∎

In Sections 3 and 4, we will prove that $T_0, T_1, \ldots, T_{n-1}$ generated by Algorithm 1 are

Algorithm 2

for each processor i $(0 \leq i \leq n)$ **do in parallel**
$son \leftarrow f_i(r)$;
$T_i \leftarrow \{(r, son)\}$;
$S \leftarrow \{son\}$;
for $m = i + 1$ to $i + n - 1$ **do**
$S' \leftarrow \emptyset$;
for each vertex $v \in S$ **do**
$u \leftarrow f_{m \bmod n}(v)$;
$T_i \leftarrow T_i \cup \{(v, u)\}$;
$S' = S' \cup \{u\}$;
endfor
$S \leftarrow S \cup S'$;
endfor
for each vertex $v \in S \setminus \{son\}$ **do**
$T_i \leftarrow T_i \cup \{(v, f_i(v))\}$;
endfor
end for

n ISTs rooted at r for LTQ_n and Q_n, respectively. Do notice that for Q_n and LTQ_n,

$$f_{n-1}^{-1} = f_{n-1},\ f_{n-2}^{-1} = f_{n-2},\ \ldots,\ f_0^{-1} = f_0.$$

Thus in the remaining discussion, we will simply write f_i instead of f_i^{-1}. The following definitions are crucial for the subsequent proofs.

Definition 4. *Consider arranging the elements* $0, 1, \ldots, n-1$ *on a circle in a clockwise manner. For all* $0 \leq i \leq n-1$*, define* O_i *to be an ordered set*

$$O_i = \{i, i-1, i-2, \ldots, i-n+1\}.$$

Here $i - k$ *means* $(i-k) \bmod n$*, where* $k = 1, 2, \ldots, n-1$*.*

Notice that O_i can be viewed as the ordered set formed by taking the elements out from the circle in a counterclockwise manner by letting i to be the first element. For example, if $n = 6$, then $O_0 = \{0, 5, 4, 3, 2, 1\}$, $O_1 = \{1, 0, 5, 4, 3, 2\}$, $O_2 = \{2, 1, 0, 5, 4, 3\}$, $O_3 = \{3, 2, 1, 0, 5, 4\}$, $O_4 = \{4, 3, 2, 1, 0, 5\}$, and $O_5 = \{5, 4, 3, 2, 1, 0\}$.

Definition 5. *For all* $0 \leq i \leq n-1$ *and* $v \in V(T_i)$*, define* $C_i(v, f_i(r))$ *as follows. Recall that* $f_i(r)$ *is the son of the root in* T_i*. Let* $v = (v_{n-1}v_{n-2}\cdots v_0)_2$ *and* $f_i(r) =$

$(a_{n-1}a_{n-2}\ldots a_0)_2$. *Suppose* v *and* $f_i(r)$ *has a total of* m *different bits. Define* $C_i(v, f_i(r))$ *to be an ordered set containing all the indices of these* m *different bits, listed according to the order given in* O_i.

We give some examples for $C_i(v, f_i(r))$. Note that when $r = 0$, the son of the root in T_i is 2^i, i.e., $f_i(r) = 2^i$. Suppose $n = 6$ and $v = (101011)_2$. Then $C_0(v, 2^0) = \{5, 3, 1\}$, $C_1(v, 2^1) = \{0, 5, 3\}$, $C_2(v, 2^2) = \{2, 1, 0, 5, 3\}$, $C_3(v, 2^3) = \{1, 0, 5\}$, $C_4(v, 2^4) = \{4, 3, 1, 0, 5\}$, and $C_5(v, 2^5) = \{3, 1, 0\}$.

Definition 6. *Suppose* $C_i(v, f_i(r)) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}$, $|C_i(v, f_i(r))| \geq 2$ *and* $j \notin C_i(v, f_i(r))$. *We say that* j *is between* c_u *and* c_{u-1} *with respect to* O_i *if when* $0, 1, \ldots, n-1$ *are arranged on a circle, the location of* j *on the circle is between* c_u *and* c_{u-1}. *Suppose* j *is between* c_u *and* c_{u-1} *with respect to* O_i. *Then when* j *is put into* $C_i(v, f_i(r))$, j *will be put into* $C_i(v, f_i(r))$ *according to its original position in* O_i.

Continue the above example. Then $4 \notin C_1(v, 2^1)$ and 4 is between $c_u = 5$ and $c_{u-1} = 3$ with respect to O_1; $2 \notin C_1(v, 2^1)$ and 2 is between 3 and 0 with respect to O_1; 4, 3 and 2 are not in $C_3(v, 2^3)$ and all of them are between 5 and 1 with respect to O_3. Since $O_3 = \{3, 2, 1, 0, 5, 4\}$, if we put 4 into $C_3(v, 2^3)$, then we obtain $\{1, 0, 5, 4\}$; if we put 2 into $C_3(v, 2^3)$, then we obtain $\{2, 1, 0, 5\}$.

Definition 7. *For all* $0 \leq i \leq n-1$ *and* $v \in V(T_i)$, *define* $\mathcal{P}_i(v, f_i(r))$ *to be an ordered set of all the indices of perfecting matchings used in the* $v, f_i(r)$*-path in* T_i, *listed according to the order from* v *to* $f_i(r)$.

Take LTQ_4 and Figures 4 for an example. Then $O_0 = \{0, 3, 2, 1\}$, $O_1 = \{1, 0, 3, 2\}$, $O_2 = \{2, 1, 0, 3\}$, $O_3 = \{3, 2, 1, 0\}$. Consider $r = 1$ and T_1. Then the son of the root is $f_1(1) = 3 = (0011)_2$. Now suppose $v = 6 = (0110)_2$. Then $v \in T_1$, $C_1(v, f_1(1)) = \{0, 2\}$ and $\mathcal{P}_1(v, f_1(1)) = \{1, 0, 2\}$. Moreover, the path from v to $f_1(1)$ is

$$(0110)_2 \overset{f_1^{-1}=f_1}{\longrightarrow} (0100)_2 \overset{f_0^{-1}=f_0}{\longrightarrow} (0101)_2 \overset{f_2^{-1}=f_2}{\longrightarrow} (0011)_2.$$

3 Applying our algorithm to locally twisted cubes

The purpose of this section is to prove that $T_0, T_1, \ldots, T_{n-1}$ generated by Algorithm 1 are n ISTs for the locally twisted cube. It is not difficult to see that LTQ_n is vertex-transitive when $n \leq 2$. LTQ_3 is vertex-transitive can be observed from Figure 3. We now prove that LTQ_n is not vertex-transitive for $n \geq 4$. For $n = 4$, let the $N_k(r)$ be the set of vertices that can be reached by r in at most k steps. Consider the number of vertices in $N_2(r)$ that reaches only one vertex in $N_1(r)$ and only one vertex in $N_3(r)$. For $r = 0$, there is only one such vertex; however, for $r = 1$, there are two such vertices. Thus LTQ_4 is not vertex-transitive. For $n \geq 5$, LTQ_n is not vertex-transitive since the BFS tree with root 0 is of height $\lceil \frac{n+3}{2} \rceil$ while the BFS tree with root 1 is of height $\lceil \frac{n+1}{2} \rceil$.

We say that two vertices $u, v \in V(G)$ are *symmetric* if there is a bijection $h : V(G) \rightarrow V(G)$ such that $h(u) = v$ and $(x, y) \in E(G)$ if and only if $(h(x), h(y)) \in E(G)$. A graph G satisfies the *odd-even-transitive property* if each pair of odd-numbered vertices are symmetric and each pair of even-numbered vertices are also symmetric.

We now prove that the locally twisted cube satisfies the odd-even-transitive property. Based on this property, we assume without loss of generality that $r = 0$ or $r = 1$ as the common root. Then, we will prove that $T_0, T_1, \ldots, T_{n-1}$ generated by Algorithm 1 are n ISTs for the locally twisted cube.

Theorem 8. *The locally twisted cube LTQ_n satisfies the odd-even-transitive property.*

Proof. It suffices to prove that (i) if v is an odd-numbered vertex and $v \neq 1$, then v and 1 are symmetric, and (ii) if v is an even-numbered vertex and $v \neq 0$, then v and 0 are symmetric. Let $\mathcal{F} = \{f_{n-1}, f_{n-2}, \ldots, f_0\}$ be defined by equation (2). Then each edge in LTQ_n is of the form $(u, f_i(u))$ for some $f_i \in \mathcal{F}$.

First consider (i). Let $v = (v_{n-1}v_{n-2} \cdots v_0)_2 \in V(LTQ_n)$ be an odd-numbered vertex and $v \neq 1$. Define a function h_1 as follows:

$$h_1(u) = v \oplus u \oplus 1 \text{ for all } u = (u_{n-1}u_{n-2} \cdots u_0)_2 \in V(LTQ_n).$$

It is not difficult to see that h_1 is a bijection from $V(LTQ_n)$ to $V(LTQ_n)$. Let $(u, f_i(u)) \in$

$E(LTQ_n)$. Then

$$h_1(u) = (v_{n-1} \oplus u_{n-1} \ v_{n-2} \oplus u_{n-2} \ \cdots \ v_1 \oplus u_1 \ u_0)_2$$

and

$$h_1(f_i(u)) = \begin{cases} (v_{n-1} \oplus u_{n-1} \ v_{n-2} \oplus u_{n-2} \ \cdots \ v_1 \oplus u_1 \ 1 \oplus \overline{u}_0)_2 & \text{if } i = 0 \\ (v_{n-1} \oplus u_{n-1} \ v_{n-2} \oplus u_{n-2} \ \cdots \ v_2 \oplus u_2 \ v_1 \oplus \overline{u}_1 \ 1 \oplus u_0)_2 & \text{if } i = 1 \end{cases}$$

and if $2 \leq i \leq n-1$, then

$$h_1(f_i(u)) = (v_{n-1} \oplus u_{n-1} \ v_{n-2} \oplus u_{n-2} \cdots v_{i+1} \oplus u_{i+1} \ v_i \oplus \overline{u}_i \ (v_{i-1} \oplus u_{i-1} \oplus u_0) \ v_{i-2} \oplus u_{i-2} \cdots v_1 \oplus u_1 \ u_0)_2.$$

Note that $v_i \oplus \overline{u}_i = \overline{v_i \oplus u_i}$ no matter $u_i = v_1$ or $u_i \neq v_i$. Therefore

$$h_1(f_i(u)) = f_i(h_1(u))$$

and hence $(h_1(u), h_1(f_i(u))) \in E(LTQ_n)$.

Now consider (ii). Let $v = (v_{n-1} v_{n-2} \cdots v_0)_2 \in V(LTQ_n)$ be an even-numbered vertex and $v \neq 0$. Define a function h_0 as follows:

$$h_0(u) = v \oplus u \text{ for all } u = (u_{n-1} u_{n-2} \cdots u_0)_2 \in V(LTQ_n).$$

It is not difficult to see that h_0 is a bijection from $V(LTQ_n)$ to $V(LTQ_n)$. Let $(u, f_i(u)) \in E(LTQ_n)$. Then

$$h_0(u) = (v_{n-1} \oplus u_{n-1} \ v_{n-2} \oplus u_{n-2} \ \cdots \ v_1 \oplus u_1 \ u_0)_2$$

and

$$h_0(f_i(u)) = \begin{cases} (v_{n-1} \oplus u_{n-1} \ v_{n-2} \oplus u_{n-2} \ \cdots \ v_1 \oplus u_1 \ \overline{u}_0)_2 & \text{if } i = 0 \\ (v_{n-1} \oplus u_{n-1} \ v_{n-2} \oplus u_{n-2} \ \cdots \ v_2 \oplus u_2 \ v_1 \oplus \overline{u}_1 \ u_0)_2 & \text{if } i = 1 \end{cases}$$

and if $2 \leq i \leq n-1$, then

$$h_0(f_i(u)) = (v_{n-1} \oplus u_{n-1} \ v_{n-2} \oplus u_{n-2} \cdots v_{k+1} \oplus u_{k+1} \ v_k \oplus \overline{u}_k \ (v_{k-1} \oplus u_{k-1} \oplus u_0) \ v_{k-2} \oplus u_{k-2} \cdots v_1 \oplus u_1 \ u_0)_2.$$

Again, $v_i \oplus \overline{u}_i = \overline{v_i \oplus u_i}$ no matter $u_i = v_1$ or $u_i \neq v_i$. Therefore

$$h_0(f_i(u)) = f_i(h_0(u))$$

and hence $(h_0(u), h_0(f_i(u))) \in E(LTQ_n)$. ∎

By Theorem 8, we assume without loss of generality that $r = 0$ or $r = 1$ as the common root. In subsections 3.1 and 3.2, we will prove that $T_0, T_1, \ldots, T_{n-1}$ generated by Algorithm 1 are n ISTs rooted at $r = 0$ and $r = 1$ for LTQ_n, respectively. For convenience, in the remaining discussion, define $I(a, b)$, where $a \geq b$, to be an ordered sequence such that

$$I(a,b) = \begin{cases} a, a-1, \ldots, b+1 & \text{if } a > b, \\ a & \text{if } a = b. \end{cases}$$

3.1 Vertex 0 as the common root

Throughout this subsection, let $T_0, T_1, \ldots, T_{n-1}$ be the output of Algorithm 1 when the input is the $\mathcal{F}$ of LTQ_n and the root is $r = 0$. The purpose of this subsection is to prove that $T_0, T_1, \ldots, T_{n-1}$ are n ISTs rooted at $r = 0$ for LTQ_n.

Lemma 9. *$T_0, T_1, \ldots, T_{n-1}$ are n spanning trees rooted at r for LTQ_n when $r = 0$.*

Proof. It suffices to prove that each T_i $(0 \leq i \leq n-1)$ is a spanning tree rooted at $r = 0$. Consider the set S used in line 6 in the algorithm. From the inner for-loop, we know that Algorithm 1 uses vertices in S to generate edges in T_i and each $v \in S$ generates exactly one edge $(u, v) \in T_i$, where $u = f_{m \bmod n}(v)$. We now claim that:

Claim: At the start of the k-th iteration of the outer for-loop, $|S| = 2^{k-1}$.

Proof of the claim. This claim is true when $k = 1$ since line 3 sets $S = \{son\}$ and hence $|S| = 1 = 2^0$. We now prove that if this claim is true before the k-th iteration of the outer for-loop, then it remains true before the next iteration. There are two cases.

Case 1: $k \in \{1, 2, \ldots, n-1\}$. Set $t = (i + k) \bmod n$ for easy writing. The k-th outer for-loop uses the perfect matching f_t to generate exactly one edge $(u, v) \in T_i$ for each

$v \in S$. Notice that the t-th bit of each vertex $v \in S$ is 0 and the t-th bit of each vertex in S' is 1. Therefore $S \cap S' = \emptyset$ before the execution of line 11. Thus at the start of the next iteration of the outer for-loop, $|S| = 2^k$.

Case 2: $k = n$. The n-th outer for-loop uses the perfect matching f_i to generate exactly one edge $(u, v) \in T_i$ for each each $v \in S$. Notice that the i-th bit of each vertex $v \in S$ is 1 and the i-th bit of each vertex in S' is 0. Therefore $S \cap S' = \emptyset$ before the execution of line 11. Thus at the start of the next iteration of the outer for-loop, $|S| = 2^k$.

∎

From the above, when the outer for-loop terminates, $k = n+1$ and $|S| = 2^n$; therefore T_i is a spanning subgraph. Also, at the end of the k-th iteration of the outer for-loop, $|S| = 2^{k-1}$ new edges are generated; thus T_i has a total of $2^0 + 2^1 + \cdots + 2^{n-1} = 2^n - 1$ edges. T_i is connected since each newly generated edge in Algorithm 1 is incident to an edge that is already generated. Thus T_i is a spanning tree rooted at $r = 0$. ∎

When $r = 0$, the son of the root in T_i is $f_i(0)$ and

$$f_i(0) = 2^i.$$

For any $v \in V(T_i) \setminus \{0, f_i(0)\}$, the $v, f_i(0)$-path in T_i can be determined by $\mathcal{P}_i(v, f_i(0))$, which can be determined by the ordered set

$$C_i(v, f_i(0)) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}$$

as follows. Suppose $v = (v_{n-1}v_{n-2}\cdots v_0)_2$. When $v_0 = 0$, since $r = 0$, we have

$$\mathcal{P}_i(v, f_i(0)) = \begin{cases} C_i(v, f_i(0)) & \text{if } i = 0, \\ \{c_{m-1} = 0, I(c_{m-2}, c_{m-3}), \ldots, I(c_3, c_2), I(c_1, c_0)\} & \text{if } i \neq 0 \text{ and } m-1 \text{ is even}, \\ \{c_{m-1} = 0, I(c_{m-2}, c_{m-3}), \ldots, I(c_2, c_1), I(c_0, 0)\} & \text{if } i \neq 0 \text{ and } m-1 \text{ is odd}. \end{cases} \quad (3)$$

When $v_0 = 1$, since $r = 0$, the set $C_i(v, f_i(0))$ must contain the value 0 if $i \neq 0$; so we assume $c_e = 0$ if $i \neq 0$. Thus when $r = 0$ and $v_0 = 1$,

$$\mathcal{P}_i(v, f_i(0)) = \begin{cases} \{I(c_{m-1}, c_{m-2}), I(c_{m-3}, c_{m-4}), \ldots, I(c_1, c_0)\} & \text{if } i = 0,\ m \text{ is even}, \\ \{I(c_{m-1}, c_{m-2}), I(c_{m-3}, c_{m-4}), \ldots, I(c_2, c_1), I(c_0, 0)\} & \text{if } i = 0,\ m \text{ is odd}, \\ \{I(c_{m-1}, c_{m-2}), I(c_{m-3}, c_{m-4}), \ldots, I(c_{e+2}, c_{e+1}), c_e, c_{e-1}, \ldots, c_0\} & \text{if } j \neq 0,\ m-e \text{ is odd}, \\ \{I(c_{m-1}, c_{m-2}), I(c_{m-3}, c_{m-4}), \ldots, I(c_{e+1}, 0), c_e, c_{e-1}, \ldots, c_0\} & \text{if } i \neq 0,\ m-e \text{ is even}. \end{cases} \quad (4)$$

In the following, we give some examples for $\mathcal{P}_i(v, f_i(0))$. Consider LTQ_4. Then $f_0(0)=2^0=1$, $f_1(0)=2^1=2$ and $f_2(0)=2^2=4$. Thus the son of the root in T_0 is 1, in T_1 is 2 and in T_2 is 4. For $v = (1010)_2 \in T_0$, we have $C_0(v, 1) = \{0, 3, 1\}$ and $\mathcal{P}_0(v, 1) = \{0, I(3, 1)\} = \{0, 3, 2\}$; so the $v, 1$-path in T_0 is

$$(1010)_2 \overset{f_0^{-1}=f_0}{\longrightarrow} (1011)_2 \overset{f_3^{-1}=f_3}{\longrightarrow} (0111)_2 \overset{f_2^{-1}=f_2}{\longrightarrow} (0001)_2.$$

For $v = (1100)_2 \in T_1$, we have $C_1(v, 2) = \{1, 3, 2\}$ and $\mathcal{P}_1(v, 2) = \{1, 3, 2\}$; so the $v, 2$-path in T_1 is

$$(1100)_2 \overset{f_1^{-1}=f_1}{\longrightarrow} (1110)_2 \overset{f_3^{-1}=f_3}{\longrightarrow} (0110)_2 \overset{f_2^{-1}=f_2}{\longrightarrow} (0010)_2.$$

For $v = (0001)_2 \in T_2$, we have $C_2(v, 4) = \{2, 0\}$ and $\mathcal{P}_2(v, 4) = \{I(2, 0), 0\} = \{2, 1, 0\}$; so the $v, 4$-path in T_2 is

$$(0001)_2 \overset{f_2^{-1}=f_2}{\longrightarrow} (0111)_2 \overset{f_1^{-1}=f_1}{\longrightarrow} (0101)_2 \overset{f_0^{-1}=f_0}{\longrightarrow} (0100)_2.$$

Lemma 10. *$T_0, T_1, \ldots, T_{n-1}$ are n vertex-independent trees rooted at r for LTQ_n when $r = 0$.*

Proof. It suffices to prove that any two T_i and T_j with $0 \leq i < j \leq n-1$ are vertex-independent, i.e., for each $v \in V(LTQ_n)$, the r, v-path in T_i and the r, v-path in T_j are internally vertex-disjoint. The son of the root in T_i is $f_i(r)$ and in T_j is $f_j(r)$. Let $v = (v_{n-1}v_{n-2}\cdots v_0)_2$ be an arbitrary vertex in LTQ_n. In the following, we assume $v \notin \{r, f_i(r), f_j(r)\}$ since if $v \in \{r, f_i(r), f_j(r)\}$, then the r, v-path in T_i and the r, v-path in T_j are clearly internally vertex-disjoint.

Since $f_i(r) \neq f_j(r)$, the r, v-path in T_i and the r, v-path in T_j are internally vertex-disjoint if and only if the $v, f_i(r)$-path in T_i and the $v, f_j(r)$-path in T_j are internally vertex-disjoint. In the following, we will only prove that the $v, f_i(r)$-path in T_i and the $v, f_j(r)$-path in T_j are internally vertex-disjoint. Let V_1 be an ordered set that contains the internal vertices of the $v, f_i(r)$-path in T_i listed from v to $f_i(r)$. Let V_2 be an ordered set that contains the internal vertices of the $v, f_j(r)$-path in T_j listed from v to $f_j(r)$. We now claim that:

Claim: $V_1 \cap V_2 = \emptyset$.

Proof of the claim. Suppose this claim is not true and there exists a vertex $a \in V_1 \cap V_2$. Recall that $f_i(0) = 2^i$ and $f_j(0) = 2^j$. Let

$$C_i(v, 2^i) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}. \tag{5}$$

There are four cases.

Case 1: $v_i = 1$ and $v_j = 1$. Then there must exist a u such that $c_u = j$. Thus

$$C_j(v, 2^j) = \{c_{u-1}, c_{u-2}, \ldots, c_0, i, c_{m-1}, c_{m-2}, \ldots, c_{u+1}\}. \tag{6}$$

By (3) and (4) and (5), c_{m-1} is the first element in $\mathcal{P}_i(v, 2^i)$. Let $x \in V_1$. Then the (c_{m-1})-th bit of x is $v_{c_{m-1}}$ only when (i) $(c_{m-1}+1) \in \mathcal{P}_i(v, 2^i)$, (ii) $c_{m-1}+1 \geq 2$ and (iii) there exists $q = (q_{n-1}q_{n-2}\cdots q_0)_2 \in V_1$ such that $x = f_{c_{m-1}+1}(q)$ and $q_0 = 1$. We now prove that (i), (ii) and (iii) will not occur simultaneously; hence for all $x \in V_1$, the (c_{m-1})-th bit of x is $\overline{v}_{c_{m-1}}$. If $|C_i(v, 2^i)| = 1$, then (i) can not occur. Suppose $|C_i(v, 2^i)| \geq 2$ and both (i) and (iii) occur; that is, there exists $q = (q_{n-1}q_{n-2}\cdots q_0)_2 \in V_1$ such that $x = f_{c_{m-1}+1}(q)$ and $q_0 = 1$. By (5), $c_{m-1}+1$ is the last element in $\mathcal{P}_i(v, 2^i)$. Since $q_0 = 1$, $I(c_0, 0) \subseteq \mathcal{P}_i(v, 2^i)$. By Lemma 3 and by the fact that $I(c_0, 0) = \{c_0, c_0 - 1, \ldots, 1\}$, we have $c_{m-1} + 1 = 1$; thus (ii) does not occur and consequently the (c_{m-1})-th bit of all the vertices in V_1 is $\overline{v}_{c_{m-1}}$. Since $v_i = 1$, the i-th bit of all the vertices in V_1 is 1. By (3) and (4) and (6), the (c_{m-1})-th bit of those vertices in V_2 with the i-th bit being 1 is $v_{c_{m-1}}$. Thus $V_1 \cap V_2 = \emptyset$.

Case 2: $v_i = 0$ and $v_j = 0$. Then $c_{m-1} = i$. If $|C_i(v, 2^i)| = 1$, then $C_i(v, 2^i) = \{i\}$, which implies that $v = 0$; this contradicts with the assumption that $v \neq 0$. Thus $|C_i(v, 2^i)| \geq 2$ and there must exist a u such that j is between c_u and c_{u-1} with respect to O_i. Thus

$$C_j(v, 2^j) = \begin{cases} \{j, c_{u-1}, c_{u-2}, \ldots, c_0, c_{m-2}, c_{m-3}, \ldots, c_{u+1}, c_u\} & \text{if } u \neq 0, \\ \{j, c_{u-1}, c_{u-2}, \ldots, c_0, c_{m-2}, c_{m-3}, \ldots, c_{u+1}\} & \text{if } u = 0. \end{cases} \tag{7}$$

By (3) and (4) and (5), the i-th bit of all vertices in V_1 is 1. By (3) and (4) and (7), the j-th bit of all vertex in V_2 is 1. Suppose $V_1 \cap V_2 \neq \emptyset$ and $a \in V_1 \cap V_2$. Then the i-th bit and the j-th bit of a are both 1. When $I(c_u, c_{u-1}) \nsubseteq \mathcal{P}_i(v, 2^i)$, each vertex in V_1 has its j-th bit to be 0. When $I(c_0, c_{m-2}) \nsubseteq \mathcal{P}_j(v, 2^j)$, each vertex in V_2 has its i-th bit to be 0. Thus the existence of a implies that $I(c_u, c_{u-1}) \subseteq \mathcal{P}_i(v, 2^i)$ and $I(c_0, c_{m-2}) \subseteq \mathcal{P}_j(v, 2^j)$. Note that $I(c_u, c_{u-1}) \subseteq \mathcal{P}_i(v, 2^i)$ implies that $i = 0$ and hence $v_0 = 0$ (since case 2 requires $v_i = 0$). However, $I(c_0, c_{m-2}) \subseteq \mathcal{P}_j(v, 2^j)$ implies $v_0 = 1$, which contradicts with $v_0 = 0$. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

Case 3: $v_i = 0$ and $v_j = 1$. Then $c_{m-1} = i$ and there must exist a u such that $c_u = j$. If $|C_i(v, 2^i)| = 1$, then $C_j(v, 2^i) = \emptyset$, which implies that $v = 2^j$; this contradicts with the assumption that $v \neq 2^j$. Thus

$$C_j(v, 2^j) = \{c_{u-1}, c_{u-2}, \ldots, c_0, c_{m-2}, c_{m-3}, \ldots, c_{u+1}\}. \tag{8}$$

By (3) and (4) and (5), the i-th bit of all vertices in V_1 is 1. Suppose $V_1 \cap V_2 \neq \emptyset$ and $a \in V_1 \cap V_2$. Then the i-th bit of a is 1. When $I(c_0, c_{m-2}) \nsubseteq \mathcal{P}_j(v, 2^j)$, each vertex in V_2 has its i-th bit to be 0. Thus the existence of a implies that $I(c_0, c_{m-2}) \subseteq \mathcal{P}_j(v, 2^j)$ which further implies $v_0 = 1$. Since $I(c_0, c_{m-2}) \subseteq \mathcal{P}_j(v, 2^j)$, V_2 has only one vertex $x = (x_{n-1}x_{n-2}\cdots x_0)_2$ such that $x_i = 1$ and $x = f_{i+1}(q)$ for some $q \in V_2$. The existence of a implies that $x = a$. Since $v_0 = 1$, $\mathcal{P}_i(v, 2^i)$ starts with $I(i, c_{m-2})$, i.e., $\mathcal{P}_i(v, 2^i)$ is of the form $\{I(i, c_{m-2}), \ldots\}$. By (4), c_{m-3} is the first element after $I(i, c_{m-2})$ in $\mathcal{P}_i(v, 2^i)$. Recall that $\mathcal{P}_i(v, 2^i)$ is an ordered set of all the indices of perfecting matchings used in the $v, 2^i$-path in T_i listed according to the order from v to 2^i. Thus the first vertex in V_1 can be obtained by applying the first perfect matching obtained from the first element in $\mathcal{P}(v, 2^i)$ to v, the second vertex in V_1 can be obtained by applying the second perfect matching obtained from the second element in $\mathcal{P}(v, 2^i)$ to the first vertex in V_1, and so on. Thus we can partition V_1 into $V_{1,a}$ and $V_{1,b}$ such that $V_{1,a}$ consists of those vertices in V_1 before $f_{c_{m-3}}$ is applied and $V_{1,b} = V_1 - V_{1,a}$. Let $y = (y_{n-1}y_{n-2}\cdots y_0)_2$ be an arbitrary vertex in $V_{1,a}$. Then bits $y_i y_{i-1}\cdots y_{c_{m-2}}$ are different from $v_i v_{i-1}\cdots v_{c_{m-2}}$ in exactly two bits. However, bits $x_i x_{i-1}\cdots x_{c_{m-2}}$ are identical to $v_i v_{i-1}\cdots v_{c_{m-2}}$. Thus $x \notin V_{1,a}$. On the other hand, $x_{c_{m-3}} = v_{c_{m-3}}$ but the (c_{m-3})-th bit of all the vertices in $V_{1,b}$ is $\overline{v}_{c_{m-3}}$; thus $x \notin V_{1,b}$. Since $x \notin V_{1,a}$ and $x \notin V_{1,b}$, we have $x \notin V_1$. Since $x = a$, it follows that $a \notin V_1$.

Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

Case 4: $v_i = 1$ and $v_j = 0$. Then there must exist a u such that j is between c_u and c_{u-1} with respect to O_i. Thus

$$C_j(v,2^j) = \begin{cases} \{j, i, c_{u-1}, c_{u-2}, \ldots, c_0, c_{m-1}, c_{m-2}, \ldots, c_u\} & \text{if } i \text{ is between } c_u \text{ and } c_{u-1} \text{ respect to } O_i, \\ \{j, c_{u-1}, c_{u-2}, \ldots, c_0, i, c_{m-1}, c_{m-2}, \ldots, c_u\} & \text{if otherwise.} \end{cases} \quad (9)$$

By (3) and (4) and (9), the j-th bit of all vertices in V_2 is 1. Since $v_i = 1$, the i-th bit of all the vertices in V_1 is 1. Suppose $V_1 \cap V_2 \neq \emptyset$ and $a \in V_1 \cap V_2$. Then the i-th bit and the j-th bit of a are both 1. By (9), case 4 consists of two subcases. In each subcase, we will prove that no such a exists. Since a does not exist, $V_1 \cap V_2 = \emptyset$.

Subcase 4.1: i is between c_u and c_{u-1} with respect to O_i. Then V_2 has only one vertex $f_j(v)$ with its i-th bit and j-th bit both being 1. By (3) and (4) and (5), c_{m-1} is the first element in $\mathcal{P}_i(v, 2^i)$. Thus the (c_{m-1})-th bit of those vertices in V_1 with the j-th bit being 1 is $\overline{v}_{c_{m-1}}$. However, by (3) and (4) and (9), the (c_{m-1})-th bit of $f_j(v)$ is $v_{c_{m-1}}$. Thus no such a exists.

Subcase 4.2: i is not between c_u and c_{u-1} with respect to O_i. If $|C_i(v, 2^i)| = 1$, then $C_i(v, 2^i) = \{c_0\}$; since $v_j = 0$, we have $c_0 \neq j$, which implies that each vertex in V_1 has its j-th bit to be 0 and consequently no such a exists. Now suppose $|C_i(v, 2^i)| \geq 2$. Then when $I(c_u, c_{u-1}) \nsubseteq \mathcal{P}_i(v, 2^i)$, each vertex in V_1 has its j-th bit to be 0. Thus the existence of a implies that $I(c_u, c_{u-1}) \subseteq \mathcal{P}_i(v, 2^i)$. Since $I(c_u, c_{u-1}) \subseteq \mathcal{P}_i(v, 2^i)$, V_1 has only one vertex $x = (x_{n-1}x_{n-2} \cdots x_0)_2$ such that $x_j = 1$ and $x = f_{j+1}(q)$ for some $q \in V_1$. The existence of a implies that $x = a$. By (3) and (4) and (9), the (c_{m-1})-th bit of those vertices in V_2 with the i-th bit being 1 is $\overline{v}_{c_{m-1}}$. However, the $x_{c_{m-1}} = v_{c_{m-1}}$. So if $x \in V_1$, $x \notin V_2$. Then, by (3) and (4) and (5), the j-th bit of all the vertices in $V_1 \setminus \{x\}$ is 0. By (3) and (4) and (9), the j-th bit of all the vertices in V_2 is 1. Thus no such a exists. ∎

Since $V_1 \cap V_2 = \emptyset$, we have this lemma. ∎

Theorem 11. *$T_0, T_1, \ldots, T_{n-1}$ are n n ISTs rooted at r for LTQ_n when $r = 0$.*

Proof. This theorem follows from Lemmas 9 and 10. ∎

3.2 Vertex 1 as the common root

Throughout this subsection, let $T_0, T_1, \ldots, T_{n-1}$ be the output of Algorithm 1 when the input is the $\mathcal{F}$ of LTQ_n and the root is $r = 1$. The purpose of this subsection is to prove that $T_0, T_1, \ldots, T_{n-1}$ are n ISTs rooted at $r = 1$ for LTQ_n. For $S \subseteq V(LTQ_n)$, define $f_i(S)$ to be

$$f_i(S) = \{f_i(v) \mid \text{for all } v \in S\}.$$

This definition will be used in the following proofs.

Lemma 12. *$T_0, T_1, \ldots, T_{n-1}$ are n spanning trees rooted at r for LTQ_n when $r = 1$.*

Proof. The proof of this lemma is similar to that of Lemma 9 except that $r = 0$ is replaced by $r = 1$ and the proof of the claim is modified as follows.

Proof of the claim. This claim is true when $k = 1$ since line 3 sets $S = \{son\}$ and hence $|S| = 1 = 2^0$. We now prove that if this claim is true before the k-th iteration of the outer for-loop, then it remains true before the next iteration. According to which T_i is considered, there are three possibilities.

1. Suppose T_0 is considered. Then $i = 0$ and there are two cases.

Case 1: $k \in \{1, 2, \ldots, n-1\}$. The proof of this case is the same as Case 1 in Lemma 9.

Case 2: $k = n$. The proof of this case is the same as Case 2 in Lemma 9 except that: the i-th bit of each vertex $v \in S$ is 0 and the i-th bit of each vertex in S' is 1.

2. Suppose T_{n-1} is considered. Then $i = n-1$ and there are two cases.

Case 1: $k \in \{1, 2, \ldots, n-1\}$. The proof of this case is the same as Case 1 in Lemma 9 except that: when $k = n-1$, the $(n-2)$-th bit of each vertex $v \in S$ is 1 and the $(n-2)$-th bit of each vertex in S' is 0.

Case 2: $k = n$. The proof of this case is the same as Case 2 in Lemma 9.

3. Suppose T_i is considered, where $i \in \{1, 2, \ldots, n-2\}$. Then there are two cases.

Case 1: $k \in \{1, 2, \ldots, n-1\}$. The proof of this case is the same as Case 1 in Lemma 9 except that: when $k = n-1$, the $(n-2)$-th bit of each vertex $v \in S$ is 1 and the $(n-2)$-th bit of each vertex in S' is 0.

Case 2: $k = n$. This is the last (the n-th) iteration of the outer for-loop of Algorithm 1. Before the n-th iteration of the outer for-loop, $|S| = 2^{n-1}$ and a total of $2^0 + 2^1 + \cdots + 2^{n-2} = 2^{n-1} - 1$ edges have been put into T_i; these edges form a connected subgraph since each newly generated edge in Algorithm 1 is incident to an edge that is already generated. Thus S induces a tree. Partition S into S^0 and S^1 such that

$$S^0 = \{\text{all the vertices in the subtree rooted at } f_{i+1}(f_i(1))\} \quad \text{and} \quad S^1 = S \setminus S^0.$$

See Figure 5 as an illustration.

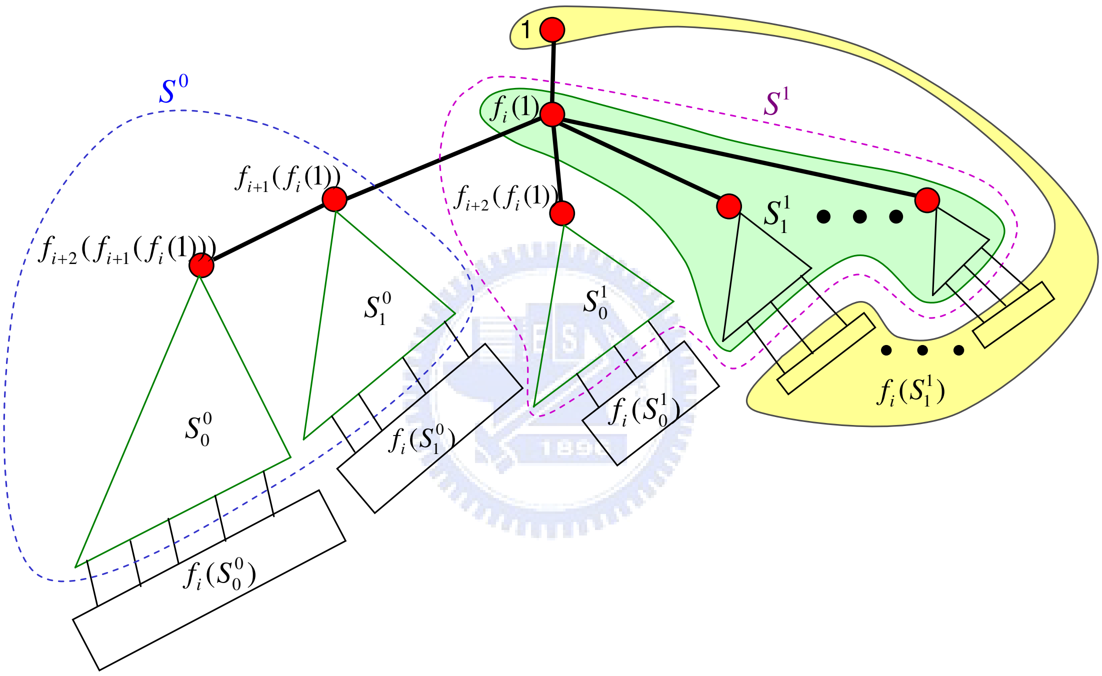

Figure 5: An illustration for the proof of Lemma 12.

By (2) and by Lemma 3, we have: (i) the i-th bit of all the vertices in S^0 is 0 and hence the i-th bit of all the vertices in $f_i(S^0)$ is 1, and (ii) the i-th bit of all the vertices in S^1 is 1 and hence the i-th bit of all the vertices in $f_i(S^1)$ is 0. Notice that

$$S' = f_i(S^0) \cup f_i(S^1).$$

By (i) and (ii), to prove that $S \cap S' = \emptyset$, it suffices to prove that

$$S^0 \cap f_i(S^1) = \emptyset \quad \text{and} \quad S^1 \cap f_i(S^0) = \emptyset. \tag{10}$$

Suppose $i = n–2$. Then the $(n–1)$-bit of all the vertices in S^0 and $f_{n-2}(S^0)$ is 1; however, the $(n–1)$-bit of all the vertices in S^1 and $f_{n-2}(S^1)$ is 0. Thus when $i = n–2$, $S^0 \cap f_{n-2}(S^1) = \emptyset$ and $S^1 \cap f_{n-2}(S^0) = \emptyset$. Now suppose $i \in \{1, 2, \ldots, n-3\}$. Partition S^0 into S^0_0 and S^0_1 such that

$$S^0_0 = \{\text{all the vertices in the subtree rooted at } f_{i+2}(f_{i+1}(f_i(1)))\} \text{ and } S^0_1 = S^0 \setminus S^0_0.$$

Partition S^1 into S^1_0 and S^1_1 such that

$$S^1_0 = \{\text{all the vertices in the subtree rooted at } f_{i+2}(f_i(1))\} \text{ and } S^1_1 = S^0 \setminus S^1_0.$$

By (2) and by Lemma 3, the pair of the $(i+1)$-th and the i-th bit of all the vertices in S^0_0 and $f_i(S^1_1)$ is (0,0); in $f_i(S^0_0)$ and S^1_1 is (0,1); in S^0_1 and $f_i(S^1_0)$ is (1,0) and in $f_i(S^0_1)$ and S^1_0 is (1,1). Thus to prove (10), it suffices to prove that

$$S^0_0 \cap f_i(S^1_1) = \emptyset,\ S^1_1 \cap f_i(S^0_0) = \emptyset,\ S^1_0 \cap f_i(S^0_1) = \emptyset \text{ and } S^0_1 \cap f_i(S^1_0) = \emptyset. \tag{11}$$

For each $v = (v_{n-1}, v_{n-1}, \ldots, v_0)_2 \in V(LTQ_n)$ such that $v \neq 0$, define q to be the index so that v_q is the leftmost nonzero bit, i.e., $v_{n-1} = v_{n-2} = \cdots = v_{q+1} = 0$ and $v_q = 1$ (since $v \neq 0$, q exists). For $v = 0$, define q to be -1. By (2) and by Lemma 3, we have Table 2. We now use two claims to prove (11).

Table 2: The value of q for every vertex in the given set.

$S^0_0 \cup f_i(S^0_0)$	$S^1_1 \cup f_i(S^1_1)$	$S^1_0 \cup f_i(S^1_0)$	$S^0_1 \cup f_i(S^0_1)$
$q \geq i+2$	$q \leq i+1$ or $q \geq i+3$	$q \geq i+3$	$q = i+1$ or $q \geq i+3$

Claim A: $S^0_0 \cap f_i(S^1_1) = \emptyset$ and $S^1_1 \cap f_i(S^0_0) = \emptyset$. This claim holds since:
By Table 2, each vertex in $S^1_1 \cap f_i(S^1_1)$ with $q \leq i+1$ does not belong to $S^0_0 \cup f_i(S^0_0)$ since every vertex in $S^0_0 \cup f_i(S^0_0)$ has $q \geq i+2$. By Table 2, each vertex in $S^0_0 \cup f_i(S^0_0)$ with $q = i+2$ does not belong to $S^1_1 \cap f_i(S^1_1)$ since each vertex in $S^1_1 \cap f_i(S^1_1)$ has $q \neq i+2$. From the above, we may focus on vertices with $q = i+3$ or $q > i+3$. Note that each vertex in $S^0_0 \cup f_i(S^0_0)$ with $q = i+3$ will have its $(i+2)$-th bit to be 0; however, from Table 2, we know that each vertex in $f_i(S^1_1) \cup S^1_1$ with $q \geq i+3$ will have its $(i+2)$-th bit to be 1. Therefore, each vertex in $S^0_0 \cup f_i(S^0_0)$ with $q = i+3$ does not belong to $S^1_1 \cup f_i(S^1_1)$.

It remains to consider vertices with $q > i+3$. Note that the bit string of those bits from the q-th bit to the $(i+2)$-th bit of all the vertices in $S_0^0 \cup f_i(S_0^0)$ is one of the strings in

$$L_0 = \{\underbrace{1\overbrace{00\cdots0}^{q-i-2\ 0's}}_{q-i-1\ \text{bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-4\ 0's}11}_{q-i-1\ \text{bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-5\ 0's}101}_{q-i-1\ \text{bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-6\ 0's}1001}_{q-i-1\ \text{bits}}, \ldots, \underbrace{101\overbrace{00\cdots0}^{q-i-5\ 0's}1}_{q-i-1\ \text{bits}}, \underbrace{11\overbrace{00\cdots0}^{q-i-4\ 0's}1}_{q-i-1\ \text{bits}}\}.$$

However, the bit string of those bits from the q-th bit to the $(i+2)$-th bit of all the vertices in $S_1^1 \cup f_i(S_1^1)$ is one of the strings in

$$L_1 = \{\underbrace{1\overbrace{00\cdots0}^{q-i-3\ 0's}1}_{q-i-1\ \text{bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-4\ 0's}10}_{q-i-1\ \text{bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-5\ 0's}100}_{q-i-1\ \text{bits}}, \underbrace{1\overbrace{00\cdots0}^{q-i-6\ 0's}1000}_{q-i-1\ \text{bits}}, \ldots, \underbrace{101\overbrace{00\cdots0}^{q-i-4\ 0's}}_{q-i-1\ \text{bits}}, \underbrace{11\overbrace{00\cdots0}^{q-i-3\ 0's}}_{q-i-1\ \text{bits}}\}.$$

It is not difficult to see that $L_0 \cap L_1 = \emptyset$. Hence we have Claim A.

Claim B: $S_1^0 \cap f_i(S_0^1) = \emptyset$ and $S_0^1 \cap f_i(S_1^0) = \emptyset$. The proof of Claim B is similar to that of Claim A except that $S_0^0 \cup f_i(S_0^0)$ is replaced by $S_0^1 \cup f_i(S_0^1)$ and $S_1^1 \cup f_i(S_1^1)$ is replaced by $S_1^0 \cup f_i(S_1^0)$.

By Claims A and B, we have (11) and hence have (10). Therefore $S \cap S' = \emptyset$ before the execution of line 11. Thus at the start of the next iteration of the outer for-loop, $|S| = 2^k$. ∎

We now have this lemma. ∎

When $r = 1$, the son of the root in T_i is $f_i(1)$, where

$$f_i(1) = \begin{cases} 0 & \text{if } i \neq 0, \\ 3 & \text{if } i = 1, \\ 2^i + 2^{i-1} + 1 & \text{if } 2 \leq i \leq n-1. \end{cases} \tag{12}$$

For any $v \in V(T_i) \setminus \{1, f_i(1)\}$, the $v, f_i(1)$-path in T_i can be determined by $\mathcal{P}_i(v, f_i(1))$, which can be determined by the ordered set

$$C_i(v, f_i(1)) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}$$

as follows. Let c_{e-1} be the first (from left to right) member in $C_i(v, f_i(1))$ that is larger than i. Suppose $v = (v_{n-1}v_{n-2}\cdots v_0)_2$. When $i = 0$, since $r = 1$, we have

$$\mathcal{P}_i(v, f_i(1)) = C_i(v, f_i(1)). \tag{13}$$

When $i \neq 0$ and $v_0 = 0$, since $r = 1$, we have $c_e = 0$ and

$$\mathcal{P}_i(v, f_i(1)) = \begin{cases} \{c_{m-1}, c_{m-2}, \ldots, c_e, I(c_{e-1}, c_{e-2}), I(c_{e-3}, c_{e-4}), \ldots, I(c_1, c_0)\} & \text{if } e \text{ is even,} \\ \{c_{m-2}, c_{m-3}, \ldots, c_e, I(c_{e-1}, c_{e-2}), I(c_{e-3}, c_{e-4}), \ldots, I(c_0, i)\} & \text{if } e \text{ is odd, } c_{m-1} = i, \\ \{i, c_{m-1}, c_{m-2}, \ldots, c_e, I(c_{e-1}, c_{e-2}), I(c_{e-3}, c_{e-4}), \ldots, I(c_0, i)\} & \text{if } e \text{ is odd, } c_{m-1} \neq i. \end{cases} \quad (14)$$

When $i \neq 0$ and $v_0 = 0$, in order to obtain $\mathcal{P}_i(v, f_i(1))$ from $C_i(v, f_i(1))$, we need to define C_i^1, C_i^2 and $\zeta_i(v, f_i(1))$. Define C_i^2 to be the ordered sequence

$$C_i^2 = c_{e-1}, c_{e-2}, \ldots, c_0$$

and define C_i^1 to be the ordered sequence

$$C_i^1 = \begin{cases} c_{m-1}, c_{m-2}, \ldots, c_e & \text{if } |C_i^2| \text{ is even,} \\ i, c_{m-1}, c_{m-2}, \ldots, c_e & \text{if } |C_i^2| \text{ is odd and } c_{m-1} \neq i \\ c_{m-2}, c_{m-3}, \ldots, c_e & \text{if } |C_i^2| \text{ is odd and } c_{m-1} = i. \end{cases}$$

Defined $\zeta_i(v, f_i(1))$ to be the ordered sequence

$$\zeta_i(v, f_i(1)) = \begin{cases} \{C_i^1, C_i^2\} & \text{if } |C_i^1| \text{ is even and } |C_i^2| \text{ is even,} \\ \{C_i^1, C_i^2, i\} & \text{if } |C_i^1| \text{ is even and } |C_i^2| \text{ is odd,} \\ \{C_i^1, 0, C_i^2\} & \text{if } |C_i^1| \text{ is odd and } |C_i^2| \text{ is even,} \\ \{C_i^1, 0, C_i^2, i\} & \text{if } |C_i^1| \text{ is odd and } |C_i^2| \text{ is odd.} \end{cases} \quad (15)$$

Suppose

$$\zeta_i(v, f_i(1)) = \{\varsigma_u, \varsigma_{u-1}, \ldots, \varsigma_0\}.$$

Then when $i \neq 0$ and $v_0 = 1$, since $r = 1$, we have

$$\mathcal{P}_i(v, f_i(1)) = \{I(\varsigma_u, \varsigma_{u-1}), I(\varsigma_{u-2}, \varsigma_{u-3}), \ldots, I(\varsigma_1, \varsigma_0), \}. \quad (16)$$

In the following, we give some examples for $\mathcal{P}_i(v, f_i(1))$. Consider LTQ_5. Then $f_1(1) = 2^1 + 1 = 3$, $f_2(1) = 2^2 + 2^1 + 1 = 7$ and $f_3(1) = 2^3 + 2^2 + 1 = 13$. Thus the son of the root in T_1 is 3, in T_2 is 7 and in T_3 is 13. For $v = (10000)_2 \in T_1$, we have $C_1(v, 3) = \{1, 0, 4\}$ and $\mathcal{P}_1(v, 3) = \{0, I(4, 1)\} = \{0, 4, 3, 2\}$; so the $v, 3$-path in T_1 is

$$(10000)_2 \stackrel{f_0^{-1}=f_0}{\longrightarrow} (10001)_2 \stackrel{f_4^{-1}=f_4}{\longrightarrow} (01001)_2 \stackrel{f_3^{-1}=f_3}{\longrightarrow} (00101)_2 \stackrel{f_2^{-1}=f_2}{\longrightarrow} (00011)_2.$$

For $v = (11010)_2 \in T_2$, we have $C_2(v, 7) = \{2, 0, 4, 3\}$ and $\mathcal{P}_2(v, 7) = \{2, 0, I(4, 3)\} = \{2, 0, 4\}$; so the $v, 7$-path in T_2 is

$$(11010)_2 \stackrel{f_2^{-1}=f_2}{\longrightarrow} (11110)_2 \stackrel{f_0^{-1}=f_0}{\longrightarrow} (11111)_2 \stackrel{f_4^{-1}=f_4}{\longrightarrow} (00111)_2.$$

For $v = (11101)_2 \in T_3$, we have $C_3(v, 13) = \{4\}$, $C_3^2 = \{4\}$, $C_3^1 = \{3\}$, $\zeta_j(v, f_j(1)) = \{3, 0, 4, 3\}$ and $\mathcal{P}_3(v, 13) = \{I(3, 0), I(4, 3)\} = \{3, 2, 1, 4\}$; so the $v, 13$-path in T_3 is

$$(11101)_2 \overset{f_3^{-1}=f_3}{\longrightarrow} (10001)_2 \overset{f_2^{-1}=f_2}{\longrightarrow} (10111)_2 \overset{f_1^{-1}=f_1}{\longrightarrow} (10101)_2 \overset{f_4^{-1}=f_4}{\longrightarrow} (01101)_2.$$

Lemma 13. *$T_0, T_1, \ldots, T_{n-1}$ are n vertex-independent trees rooted at r for LTQ_n when $r = 1$.*

Proof. It suffices to prove that any two T_i and T_j with $0 \leq i < j \leq n-1$ are vertex-independent. Let $v = (v_{n-1}v_{n-2}\cdots v_0)_2$ be an arbitrary vertex in LTQ_n. We assume $v \notin \{r, f_i(r), f_j(r)\}$ since if $v \in \{r, f_i(r), f_j(r)\}$, then the r, v-path in T_i and the r, v-path in T_j are clearly internally vertex-disjoint. By the same arguments used in the proof of Lemma 10, it suffices to prove that the $v, f_i(r)$-path in T_i and the $v, f_j(r)$-path in T_j are internally vertex-disjoint. Let V_1 and V_2 be defined as in Lemma 10. We now claim that:

Claim: $V_1 \cap V_2 = \emptyset$.

Proof of the claim. Suppose this claim is not true and there exists a vertex $a \in V_1 \cap V_2$. Let

$$C_i(v, f_i(1)) = \{c_{m-1}, c_{m-2}, \ldots, c_0\}. \tag{17}$$

There are four cases.

Case 1: $0 = i < j \leq n-1$. The proof of this case is divided into two parts, depending on $v_0 = 1$ or $v_0 = 0$. Suppose $v_0 = 1$. Then $0 \notin C_j(v, f_j(1))$. Thus the 0-th bit of all the vertices in V_2 is 1. By (13) and (17), 0 is the first element in $C_0(v, f_0(1))$; this implies that the 0-th bit of all the vertices in V_1 is 0. Thus $V_1 \cap V_2 = \emptyset$. Suppose $v_0 = 0$. Then $0 \notin C_0(v, f_0(1))$. Thus the 0-th bit of all the vertices in V_1 is 0; this implies that the 0-th bit of a is 0. There are two possibilities: $j = 1$ or $j > 1$.

1. $j = 1$. Note that either $1 \in \mathcal{P}_1(v, f_1(1))$ or $1 \notin \mathcal{P}_1(v, f_1(1))$. Suppose $1 \notin \mathcal{P}_1(v, f_1(1))$. Then 0 is the first element in $\mathcal{P}_1(v, f_1(1))$; this implies that the 0-th bit of all the vertices in V_2 is 1. Thus no such a exists and $V_1 \cap V_2 = \emptyset$. Suppose $1 \in \mathcal{P}_1(v, f_1(1))$. Then 1 and 0 are the first element and the second element in $\mathcal{P}_1(v, f_1(1))$, respectively. Thus the

0-th bit of all the vertices in $V_2 \setminus \{f_1(v)\}$ is 1. The existence of a implies that $f_1(v) = a$. Suppose $v_1 = 0$. Then $1 \notin C_0(v, f_0(1))$; this implies that the 1-st bit of all the vertices in V_1 is 0. However, it is obvious that the 1-st bit of $f_1(v)$ is 1. Therefore $f_1(v) \notin V_1$. Thus no such a exists and $V_1 \cap V_2 = \emptyset$. Suppose $v_1 = 1$. Since $1 \in \mathcal{P}_1(v, f_1(1))$, there must exist some $k > 1$ such that $v_k = 1$; this implies that $c_{m-1} \neq 1$. By (13) and (17), the (c_{m-1})-th bit of all the vertices in V_1 is $\overline{v}_{c_{m-1}}$. However, the (c_{m-1})-th bit of $f_1(v)$ is $v_{c_{m-1}}$. Therefore $f_1(v) \notin V_1$. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

2. $j > 1$. By (13), (14), (15), (16) and (17), we have: c_{m-1} is the first element in $C_i(v, f_i(1))$, $c_{m-1} \in C_j(v, f_j(1))$, $0 \in C_j(v, f_j(1))$, and c_{m-1} appears after 0 in the ordered set $C_j(v, f_j(1))$. Thus the (c_{m-1})-th bit of all the vertices in V_1 is $\overline{v}_{c_{m-1}}$. However, the (c_{m-1})-th bit of those vertices with the 0-th bit being 0 in V_2 is $v_{c_{m-1}}$. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

Case 2: $1 = i < j \leq n - 1$. The proof of this case is divided into two parts, depending on $v_0 = 0$ or $v_0 = 1$.

1. $v_0 = 0$. Then it is not difficult to see (by comparing the j-th and the 0-th bits of $f_j(v)$ and all the vertices in V_1) that $f_j(v) \notin V_1$. Thus a can not be $f_j(v)$. It remains to consider those vertices in $V_2 \setminus f_j(v)$. The remaining proof is further divided into two parts, depending on $v_{j-1} = 0$ or $v_{j-1} = 1$.

1.1. $v_{j-1} = 0$. Since $v_0 = 0$ and $v_{j-1} = 0$, $j - 1 \in \mathcal{P}_j(v, f_j(v))$. Since $v_0 = 0$ and $j - 1 \in \mathcal{P}_j(v, f_j(v))$, the $(j - 1)$-th bit of all the vertices in $V_2 \setminus f_j(v)$ is 1. However, the $(j - 1)$-th bit of all the vertices in V_1 is 0. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

1.2. $v_{j-1} = 1$. We claim that: the bits from v_{j-2} to v_2 are all 0, i.e., $v_{j-2} = v_{j-3} = \cdots = v_2 = 0$. Suppose this claim is not true and let k be the largest number between $j - 2$ and 2 (inclusive) such that $v_k = 1$. By (17) and (14), the $(j - 1)$-th and the k-th bits of all the vertices in $V_2 \setminus f_j(v)$ is 1 and 0, respectively. However, the $(j - 1)$-th bit of those vertices in V_1 with k-th bit being 0 is 0. Thus $v_{j-2} = v_{j-3} = \cdots = v_2 = 0$. So the 1-st bit of all the vertices in V_1 is 1 and the 1-st bit of all the vertices in $V_2 \setminus f_j(v)$ is 0. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

2. $v_0 = 1$. The proof of this part is further divided into six parts as follows.

2.1. $j = 2$, $v_1 = 1$ and $v_2 = 1$. Since $v_0 = 1$ and $v_1 = 1$ and $v_2 = 1$,

$$C_j(v, f_j(1)) = (c_{m-1}, c_{m-2}, \ldots, c_1).$$

Suppose m is even. Then

$$\mathcal{P}_i(v, f_i(1)) = \{I(c_{m-1}, c_{m-2}), \ldots, I(c_1, c_0 = 2)\}$$

and

$$\mathcal{P}_j(v, f_j(1)) = \{I(2, 0), I(c_{m-1}, c_{m-2}), \ldots, I(c_1, 2)\}.$$

By (15) and (16), the 2-nd bit of all the vertices in V_1 are 1. However, the 2-nd bit of all the vertices in V_2 are 0. Thus no such a exists and $V_1 \cap V_2 = \emptyset$. Suppose m is odd. Then

$$\mathcal{P}_i(v, f_i(1)) = \{1, I(c_{m-1}, c_{m-2}), \ldots, I(c_0, 1)\}$$

and

$$\mathcal{P}_j(v, f_j(1)) = \{I(c_{m-1}, c_{m-2}), \ldots, I(c_2, c_1)\}.$$

By (15) and (16), the 1-st bit of all the vertices in V_1 is 0. However, the 1-st bit of all the vertices in V_2 is 1. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

2.2. $j = 2$, $v_1 = 0$ and $v_2 = 1$. Since $v_0 = 1$ and $v_1 = 0$ and $v_2 = 1$, we have $c_{m-1} = 1$, $c_0 = 2$ and

$$C_j(v, f_j(1)) = \{c_{m-1}, c_{m-2}, \ldots, c_1\}.$$

Suppose $m - 1$ is odd. Then

$$\mathcal{P}_i(v, f_i(1)) = \{I(c_{m-2}, c_{m-3}), \ldots, I(c_0, 1)\}$$

and

$$\mathcal{P}_j(v, f_j(1)) = \{1, I(c_{m-2}, c_{m-3}), \ldots, I(c_2, c_1)\}.$$

By (15) and (16), the 1-st bit of all vertices in V_1 are 0. However, the 1-st bit of all vertices in V_2 is 1. Thus no such a exists and $V_1 \cap V_2 = \emptyset$. Suppose $m - 1$ is even. Then

$$\mathcal{P}_i(v, f_i(1)) = \{1, I(c_{m-2}, c_{m-3}), \ldots, I(c_1, c_0)\}$$

and

$$\mathcal{P}_j(v, f_j(1)) = \{2, 1, I(c_{m-2}, c_{m-3}), \ldots, I(c_1, 2)\}.$$

By (15) and (16), the 2-nd bit of all vertices in V_1 are 1. However, the 2-nd bit of all vertices in V_2 are 0. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

2.3. $j = 2$, $v_1 = 1$ and $v_2 = 0$ (resp., $v_1 = 0$ and $v_2 = 0$). Then

$$C_j(v, f_j(1)) = \{2, c_{m-1}, c_{m-2}, \ldots, c_0\}.$$

Suppose m (resp., $m-1$) is even. Then by (15) and (16), the 2-nd bit of all vertices in V_1. However, the 2-nd bit of all vertices in V_2 are 1. Suppose m (resp., $m-1$) is odd. Then by (15) and (16), the 1-st bit of all vertices in V_1 are 0. However, the 1-st bit of all vertices in V_2 are 1. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

2.4. $j \neq 2$ and $v_{j-1} = 0$. Then the $(j-1)$-th bit of all the vertices in V_1 are 0. However, the $(j-1)$-th bit of all the vertices in V_2 are 1. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

2.5. $j \neq 2$, $v_{j-1} = 1$ and at least one of the bits in $v_{j-2}v_{j-3}\cdots v_2$ is 1. Then there exist q such that

$$q = \max\{\ t \ \mid \ t \in C_i(v, f_i(1)),\ 1 < t < j-1\}.$$

2.5.1. Suppose $I(j,q) \nsubseteq \mathcal{P}_j(v, f_j(1))$. Then the q-th and the $(j-1)$-th bit of all the vertices in V_2 are 0 and 1, respectively; however, the $(j-1)$-th bit of those vertices in V_1 with the q-th bit being 0 is 0. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

2.5.2. Suppose $I(j,q) \subseteq \mathcal{P}_j(v, f_j(1))$. Then we partition V_2 into $V_{2,1}$ and $V_{2,2}$ such that

$$V_{2,1} = \{\text{all the vertices in } V_2 \text{ before the perfect matching } f_q \text{ is applied}\} \text{ and } V_{2,2} = V_2 \backslash V_{2,1}.$$

Consider the vertices in $V_{2,1}$. Suppose $v_j = 0$. Since $j \in I(j,q)$, we can compare the j-th bit of all vertices in V_1 and in $V_{2,1}$ to see that no such a exists and $V_1 \cap V_2 = \emptyset$. Suppose $v_j = 1$. Then the number of bits in $v_{n-1}v_{n-2}\cdots v_{j+1}$ that are 1 is odd. This implies that $c_{m-1} \neq j$. Since $c_{m-1} \neq j$, by comparing the c_{m-1}-th bit of all the vertices in V_1 and in $V_{2,1}$, we know that $V_1 \cap V_{2,1} = \emptyset$. Consider the vertices in $V_{2,2}$. Then the q-th and the

$(j-1)$-th bit of all the vertices in $V_{2,2}$ are 0 and 1, respectively. However, the $(j-1)$-th bit of those vertices in V_1 with the q-th bit being 0 is 0. Hence $V_1 \cap V_{2,2} = \emptyset$. Since $V_1 \cap V_{2,1} = \emptyset$ and $V_1 \cap V_{2,2} = \emptyset$, no such a exists and $V_1 \cap V_2 = \emptyset$.

2.6. $j \neq 2$, $v_{j-1} = 1$ and all the bits in $v_{j-2}v_{j-3}\cdots v_2$ are 0 (i.e., $v_{j-2}{=}v_{j-3}{=}\cdots{=}v_2{=}0$). For convenience, let $t(w_1, w_2)$ denote the number of bits in $v_{w_1}v_{w_1-1}\cdots v_{w_2}$ that are 1. There are three possibilities.

2.6.1. Suppose $t(n-1, i+1)$ is odd. Then $t(n-1, j)$ is even. Thus the i-th bit of all the vertices in V_2 is 0. However, the i-th bit of all the vertices in V_1 is 1. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

2.6.2. Suppose $t(n-1, i+1)$ is even and $v_j = 0$. Then $t(n-1, j+1)$ is even. Thus the j-th bit of all the vertices in V_2 is 1. However, the j-th bit of all the vertices in V_1 is 0. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

2.6.3. Suppose $t(n-1, i+1)$ is even and $v_j = 1$. Then $t(n-1, j+1)$ is odd. Thus the i-th bit of all the vertices in $V_2 \setminus \{f_j(v)\}$ is 0. However, the i-th bit of all the vertices in V_1 is 1. Since $c_{m-1} \neq j$, we can find that $f_j(v) \notin V_1$ by comparing the c_{m-1}-th bit. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

Case 3: $3 \leq i+1 = j \leq n-1$. For convenience, let t denote the number of bits in $v_{n-1}v_{n-2}\cdots v_{i+1}$ that are 1. By (13)$\sim$(17), we have the following results for t. Suppose t is odd. Then the i-th bit of all vertices in V_1 is 0 and $j \notin \mathcal{P}_j(v, f_j(1))$; however, the i-th bit of all the vertices in V_2 is 1. Suppose t is even and $v_j = 0$. Then the j-th bit of all the vertices in V_2 is 1; however, the j-th bit of all the vertices in V_1 is 0. Suppose t is even and $v_j = 1$. Then the j-th bit of all the vertices in V_2 is 0; however, the j-th bit of all the vertices in V_1 is 1. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

Case 4: $3 \leq i+1 < j \leq n-1$. The proof of this case is divided into xxx parts, depending on the values of v_{j-1} and v_{i-1}.

4.1. $v_{j-1} = 0$. Then if $j \in \mathcal{P}_i(v, f_i(1))$, then V_1 has only one vertex (say, vertex x) with its $(j-1)$-th bit being 1. By comparing from the j-th to the $(i-1)$-th bits of x with the

j-th to the $(i-1)$-th bits of each vertex in V_2, we have $x \notin V_2$. If $j \in \mathcal{P}_j(v, f_j(1))$, then $f_j(v)$ is the unique vertex in V_2 with its $(j-1)$-th bit being 0. By comparing from the j-th to the $(i-1)$-th bits of $f_j(v)$ with the j-th to the $(i-1)$-th bits of each vertex in V_1, we have $f_j(v) \notin V_1$. Then by (13)$\sim$(17), the $(j-1)$-th bit of all the vertices in $V_1 \setminus \{x\}$ is 0; however, the $(j-1)$-th bit of all the vertices in $V_2 \setminus f_j(v)$ is 1. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

4.2. $v_{i-1} = 0$. Then we can use similar arguments to prove that no such a exists and $V_1 \cap V_2 = \emptyset$.

4.3. $v_{i-1} = 1$ and $v_{j-1} = 1$. By (13)$\sim$(16), we have following the results. When $i \in C_i(v, f_i(1))$ and $v_0 = 1$, V_1 has only one vertex (say, vertex z) with its $(i-1)$-th bit being 0. By comparing the $(j-1)$-th and the $(i-1)$-th bits of z with the $(j-1)$-th and the $(i-1)$-th bits of each vertex in V_2, we have $z \notin V_2$. Thus the $(i-1)$-th bit of all the vertices in $V_1 \setminus \{z\}$ is 1. Hence the existence of a implies that the $(i-1)$-th bit of a must be 1. Partition V_2 into two $V_{2,1}$ and $V_{2,2}$ such that

$$V_{2,1} = \{\text{all the vertices in } V_2 \text{ before the perferct matching } f_i \text{ is applied}\} \text{ and } V_{2,2} = V_2 \backslash V_{2,1}.$$

Thus the $(i-1)$-th bit of all the vertices in $V_{2,1}$ is 1 and if a exist, then $a \in V_{2,1}$. We claim that:

$$\text{If } a \text{ exists, then } v_{j-2} = v_{j-3} = \cdots = v_{i+1} = 0.$$

Suppose this claim is not true. Then let q be the largest index between $j-2$ and $i+1$ (inclusive) such that $v_q = 1$. Let $y = (y_{n-1}y_{n-2}\cdots y_0)_2$ be an arbitrary vertex in $V_{2,1} \setminus \{f_j(v)\}$. Note that $f_j(v) \in V_{2,1}$ only when $j \in \mathcal{P}_j(v, f_j(1))$. Also note that $q \in \mathcal{P}_j(v, f_j(1))$. Moreover, if $j \in C_j(v, f_j(1))$, then q is the first element after j in $C_j(v, f_j(1))$; if $j \notin C_j(v, f_j(1))$, then q is the first element in $C_j(v, f_j(1))$. Since q exists, by (14)$\sim$(16), the bits $y_{j-2}y_{j-3}\cdots y_{i+1}$ will be different from the bits $v_{j-2}v_{j-3}\cdots v_{i+1}$. However, let $x = (x_{n-1}x_{n-2}\cdots x_0)_2$ be an arbitrary vertex in V_1. Then the bits $x_{j-2}x_{j-3}\cdots x_{i+1}$ are identical to the bits $v_{j-2}v_{j-3}\cdots v_{i+1}$. Thus every vertex in $V_{2,1} \setminus \{f_j(v)\}$ is not in V_1. Although $f_j(v) \in V_{2,1}$, $f_j(v)$ is not in V_1 (this can be observed by comparing the j-th bit and the bits from the $(j-2)$-th to the $(i+1)$-th bits of all the vertices in V_1 with j-th bit and the bits from the $(j-2)$-th to the $(i+1)$-th bits of $f_j(v)$). Thus $V_1 \cap V_{2,1} = \emptyset$. Since if a exists, then $a \in V_{2,1}$. Thus a does not exists and we have this claim.

By this claim, in the remaining proof, we assume $v_{i-1}=1$, $v_{j-1}=1$ and $v_{j-2}=v_{j-3}=\cdots=v_{i+1}=0$. For convenience, let t denote the number of bits in $v_{n-1}v_{n-2}\cdots v_{j+1}$ that are 1. The remaining proof is further divided into four subcases.

4.3.1. $v_i = 1$ and $v_j = 1$. Suppose t is even. Then the first member in $\mathcal{P}_j(v, f_j(1))$ is i. However, $i \notin \mathcal{P}_i(v, f_i(1))$. Thus no such a exists and $V_1 \cap V_2 = \emptyset$. Suppose t is odd. Then $j \in \mathcal{P}_j(v, f_j(1))$ and $I(j-1, i) \subset \mathcal{P}_i(v, f_i(1))$. Thus the j-th bit of all the vertices in V_2 is 0. Partition V_1 into $V_{1,1}$ and $V_{1,2}$ such that

$V_{1,1}=\{$all the vertices in V_1 before the perfect matching f_{j+1} is applied$\}$ and $V_{1,2}=V_1 \backslash V_{1,1}$.

Thus the j-th bit of all vertices in $V_{1,1}$ is 1 and the j-th bit of all vertices in $V_{1,2}$ is 0. By the fact that the j-th bit of all the vertices in V_2 is 0, to prove $V_1 \cap V_2 = \emptyset$, it suffices to prove $V_{1,2} \cap V_2 = \emptyset$. If $v_0 = 1$, then the $(j-1)$-th bit of all the vertices in V_2 is 1; however, the $(j-1)$-th bit of all the vertices in $V_{1,2}$ is 0. If $v_0 = 0$, then V_2 has only one vertex $f_j(v)$ with its $(j-1)$-th bit being 0. Obviously, either $f_j(v) = (v_{n-1}v_{n-2}\cdots v_{j+1}0v_{j-1}v_{j-2}v_{j-3}\cdots v_0)_2$ or $f_j(v) = (v_{n-1}v_{n-2}\cdots v_{j+1}0\overline{v}_{j-1}v_{j-2}v_{j-3}\cdots v_0)_2$; the former case occurs when $v_0=0$ and the latter, $v_0=1$. In either case, we have $f_j(v) \notin V_1$. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

4.3.2. $v_i = 0$ and $v_j = 0$. Suppose t is even. Then the j-th bit of all the vertices in V_2 is 1. However, the j-th bit of all the vertices in V_1 is 0. Suppose t is odd. Then the number of bits in $v_{n-1}v_{n-2}\cdots v_{i+1}$ that are 1 is even; this implies that i is the first member in $\mathcal{P}_i(v, f_i(1))$. Thus the i-th bit of all the vertices in V_2 is 0. However, the i-th bit of all the vertices in V_1 is 1. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

4.3.3. $v_i = 0$ and $v_j = 1$. Suppose t is even. Then the first member in $\mathcal{P}_j(v, f_j(1))$ is $i-1$ and the first member in $\mathcal{P}_i(v, f_i(1))$ is i. So the i-th bit of all the vertices in V_2 is 0; however, the i-th bit of all the vertices in V_1 is 1. Suppose t is odd. Define q to be the index of the leftmost nonzero bit of v. Then $q > j$. Thus the $(i-1)$-th bit of all the vertices in $V_2 \setminus \{f_j(v)\}$ is 0; however, the $(i-1)$-th bit of all the vertices in V_1 is 1. By comparing the j-th and the q-th bits of $f_j(v)$ with the j-th and the q-th bits of every vertex in V_1, we have $f_j(v) \notin V_1$. Thus no such a exists and $V_1 \cap V_2 = \emptyset$.

4.3.4. $v_i = 1$ and $v_j = 0$. If the number of those bits from v_{n-1} to v_{j+1} being 1 is even, then the j-th bit of all the vertices in V_2 is 1, but the j-th bit of all the vertices in V_1 is 0. If the number of those bits from v_{n-1} to v_{j+1} being 1 is odd, then the number of bits in $v_{n-1}v_{n-2}\cdots v_{i+1}$ that are 1 is even. Thus i is the first member of $\mathcal{P}_j(v, f_j(1))$ but $i \notin \mathcal{P}_i(v, f_j(1))$ which implies that the i-th bit of all the vertices in V_2 is 0 but the i-th bit of all the vertices in V_1 is 1. So $V_1 \cap V_2 = \emptyset$ in this case. ∎

Since $V_1 \cap V_2 = \emptyset$, we have this lemma. ∎

Theorem 14. *$T_0, T_1, \ldots, T_{n-1}$ are n ISTs rooted at r for LTQ_n when $r = 1$.*

Proof. This theorem follows from Lemmas 12 and 13. ∎

4 Applying our algorithm to hypercubes

The purpose of this section is to prove that $T_0, T_1, \ldots, T_{n-1}$ generated by Algorithm 1 are n ISTs for the hypercube. It is well-known that the hypercube is vertex-transitive. Therefore we assume without loss of generality that $r = 0$ is the common root. Throughout this section, let $T_0, T_1, \ldots, T_{n-1}$ be the output of Algorithm 1 when the input is the $\mathcal{F}$ of Q_n and the root is $r = 0$. It is not difficult to see that the hypercube has

$$\mathcal{P}_i(v, f_i(r)) = C_i(v, f_i(r)), \text{ for all } 0 \leq i \leq n-1.$$

Theorem 15. *$T_0, T_1, \ldots, T_{n-1}$ are n ISTs rooted at r for Q_n when $r = 0$.*

Proof. We first prove that $T_0, T_1, \ldots, T_{n-1}$ are spanning trees of Q_n. The proof of this part is identical to the proof of Lemma 9 except that the definition of $\mathcal{F}$ is the one for Q_n. It remains to prove that $T_0, T_1, \ldots, T_{n-1}$ are n vertex-independent trees rooted at r for Q_n when $r = 0$. Consider an arbitrary vertex $v = (v_{n-1}, v_{n-2}\cdots v_0)_2 \in V(Q_n) \setminus \{r\}$. We use the definitions of T_i, T_j, V_1, V_2 and $C_i(v, 2^i)$ in Lemma 10. Note that in each of the following four cases, $C_j(v, 2^j)$ is also the same as the one used in Lemma 10. To prove that T_i and T_j are vertex-independent, it suffices to prove that $V_1 \cap V_2 = \emptyset$ holds in each cases.

Case 1: $v_i = 1$ and $v_j = 1$. By (5), the i-th bit of all the vertices in V_1 is 1. Partition V_2 into $V_{2,1}$ and $V_{2,2}$ such that

$$V_{2,1} = \{\text{all the vertices in } V_1 \text{ before the perfect matching } f_i \text{ is applied}\} \text{ and } V_{2,2} = V_2 \backslash V_{2,1}.$$

Thus the i-th bit of all the vertices in $V_{2,1}$ is 1 and the i-th bit of all the vertices in $V_{2,2}$ is 0. By the fact that the i-th bit of all the vertices in V_1 is 1, to prove $V_1 \cap V_2 = \emptyset$, it suffices to prove $V_1 \cap V_{2,1} = \emptyset$. By (5) and (6), the (c_{m-1})-th bit of all the vertices in V_1 is $\overline{v}_{c_{m-1}}$; however, the (c_{m-1})-th bit of all the vertices in $V_{2,1}$ is $v_{c_{m-1}}$. Thus $V_1 \cap V_{2,1} = \emptyset$.

Case 2: $v_i = 0$ and $v_j = 0$. By (5), (7) and (8), the i-th bit of all the vertices in V_1 is 1; however, the i-th bit of all the vertices in V_2 is 0. Thus $V_1 \cap V_2 = \emptyset$.

Case 3: $v_i = 0$ and $v_j = 1$. The proof of this part is the same as Case 2 and we omit it.

Case 4: $v_i = 1$ and $v_j = 0$. By (5) and (9), the j-th bit of all the vertices in V_1 is 0; however, the j-th bit of all the vertices in V_2 is 1. Thus $V_1 \cap V_2 = \emptyset$.

By above four cases, $V_1 \cap V_2 = \emptyset$. Thus $T_0, T_1, \ldots, T_{n-1}$ are n vertex-independent trees. Since $T_0, T_1, \ldots, T_{n-1}$ are also spanning trees, we have this theorem. ∎

Let $N(r)$ be a vertex set containing all the neighbors of r. The following lemma has been proven in [22].

Lemma 16. [22] *Given a n-connected, n-regular graph G and a set $\mathcal{S}$ of independent spanning trees rooted at r in G. Let v be a vertex in G, $v \notin \{r\} \cup N(r)$, and $u \in N(v)$. If $|d(T_i; r, u) - d(T_i; r, v)| \leq 1$ for every $T \in \mathcal{S}$, then $\mathcal{S}$ is optimal.*

We now prove that Algorithm 1 generates an optimal solution for Q_n.

Theorem 17. *Let $\mathcal{S} = \{T_0, T_1, \ldots, T_{n-1}\}$, where $T_0, T_1, \ldots, T_{n-1}$ are renerated by Algorithm 1. Then $\mathcal{S}$ is optimal.*

Proof. Let $r = 0$, $T_i \in \mathcal{S}$, and $H(u, v)$ be the Hamming distance between vertices v and u. Let v be an arbitrary vertex in Q_n and $v \notin \{r\} \cup N(r)$. For each T_i, we will prove that

v has the property that $|d(T_i;0,u) - d(T_i;0,v)| \leq 1$, where $u \in N(v)$. It is obvious that for each vertex $a = (a_{n-1}a_{n-2}\cdots a_0)_2$, we have

$$d(T_i;0,a) = \begin{cases} H(0,a) & \text{if } a_i = 1, \\ H(0,a)+2 & \text{if } a_i = 0. \end{cases}$$

Thus if the i-th bit of v and the i-th bit of u are the same, then $|d(T_i;0,u)-d(T_i;0,v)| = 1$. On the other hand, without loss of generality, we may assume that the i-th of v is 1 and the i-th of u is 0. Since $H(0,v)=H(0,u)+1$, we have $d(T_i;0,u) = H(0,u)+2 = H(0,v)+1 = d(T_i;0,v)+1$; hence $|d(T_i;0,u) - d(T_i;0,v)| = 1$. By Lemma 16, we have this theorem. ∎

5 Concluding remarks

There are two versions for the n independent spanning trees conjecture. The vertex (edge) conjecture is that any n-connected (n-edge-connected) graph has n vertex-independent (edge-independent) spanning trees rooted at an arbitrary vertex r. It has been proven that the vertex conjecture implies the edge conjecture. In this thesis, we present an algorithm to construct n vertex-independent spanning trees rooted at any vertex for the LTQ_n. To the best of our knowledge, this is the first result to confirm the Vertex Conjecture for the locally twisted cubes. Moreover, we present the first algorithm that can construct n vertex-independent spanning trees rooted at any vertex for both the locally twisted cube and the hypercube. We believe that our algorithm can be used to construct n vertex-independent spanning trees rooted at any vertex for other variant the hypercube.

References

[1] F. Bao , Y. Funyu, Y. Hamada and Y. Igarashi , "Reliable broadcasting and secure distributing in channel networks," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E81A, pp. 796-806, 1998.

[2] F. Bao , Y. Igarashi and S. R. Ohring, "Reliable broadcasting in product networks," *Graph-theoretic Concepts in Computer Science*, vol. 1517, pp. 310-323, 1998.

[3] Y. S. Chen, C. Y. Chiang and C.Y. Chen, "Multi-node broadcasting in all-ported 3-D wormhole-routed torus using an aggregation-then-distribution strategy," *Journal of System Architecyure*, vol. 50, pp. 575-589, 2004.

[4] G. Chartrand and L. Lensniak, Graph and Digraphs, Wadsworth, Monterey, CA, 1981.

[5] S. Curran, O. Lee and X. Yu, "Finding four independent trees," *SIAM Journal on Computing*, vol. 35, pp. 1023-1058, 2006.

[6] J. Cheriyan and S.N. Maheshwari, "Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs," *Journal of Algorithms*, vol. 9, pp. 507-537,1988.

[7] J. Edmonds, "Edge-disjoint branchings," in: R. Rustin, ed., *Combinatorial Algorithms*, Courant Inst. Sci. Symp. 9 (Algorithmics Press, New York, 1973), pp. 91V96.

[8] Z. Ge and S. L. Hakimi, "Disjoint rooted spanning trees with small depths in deBruijn and Kautz graphs," *SIAM Journal on Computing*, vol. 26, pp. 79-92, 1997.

[9] A. Huck, "Independent trees in planar graphs," *Graphs and Combinatorics*, vol. 5, pp. 29-77, 1999.

[10] A. Huck, "Independent trees in graphs," *Graphs and Combinatorics*, vol. 10, pp. 29-45, 1994.

[11] T. Hasunuma and H. Nagamochi, "Independent spanning trees with small depths in iterated line digraphs," *Discrete Applied Mathematics*, vol. 110, pp. 189-211, 2001.

[12] S. Y. Hsieh and C. J. Tu, "Constructing edge-disjoint spanning trees in locally twisted cubes," *Theoretical Computer Science*, vol. 410, pp. 8-10, 2009.

[13] K. S. Hu, S. S. Yeoh, C. Y. Chen, and L. H. Hsu, "Node-pancyclicity and edge-pancyclicity of hypercube variants," *Information Processing Letters*, vol. 102 (1) pp. 1-7, 2007.

[14] Y. Iwasaki, Y. Kajiwara, K. Obokata and Y. Igarashi, "Independent spanning trees of chordal rings," *Information Processing Letters*, vol. 69, pp. 155-160, 1999.

[15] A. Itai and M. Rodeh, "The multi-tree approach to reliability in distributed networks," *Information and Computation*, vol. 79, pp. 43-59, 1988.

[16] S. Khuller, B. Schieber, "On Insependent spanning-trees," *Information Processing Letters*, vol. 42, pp. 321-323, 1992.

[17] K. Miura, D. Takahashi, S. Nakano and T. Nishizeki, "A linear-time algorithm to find four independent spanning trees in four-connected planar graphs," *International Journal of Foundations of Computer Science*, vol. 10, pp. 195-210, 1999.

[18] K. Miura, D. Takahashi, S. Nakano and T. Nishizeki, "A linear-time algorithm to find four independent spanning trees in four-connected planar graphs," *Discrete Applied Mathematics*, vol. 83, pp. 3-20, 1998.

[19] S. Nagai and S. Nakano, "A linear-time algorithm to find independent spanning trees in maximal planar graphs," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E84A, pp. 1102-1109, 2001; also appears in: *Proceedings of 26th Workshop on Graph-Theoretic Concepts in Computer Science*, WG 2000, LNCS 1928, Springer, 2000, pp. 290-301.

[20] K. Obokata, Y. Iwasaki, F. Bao and Y. Igarashi, "Independent spanning trees of product graphs," *Lecture Notes in Computer Science*, vol. 197, pp. 338-351, 1996. See also: K. Obokata, Y. Iwasaki, F. Bao, Y. Igarashi, "Independent spanning trees of product graphs and their construction," *IEICE Trans. Fundamentals of Electronics,Communications and Computer Sciences*, E79-A, pp. 1894V1903, 1996.

[21] Y. C. Tseng, S. Y. Wang and C. W. Ho, "Efficient broadcasting in wormhole-routed multicomputers: A network-partitioning approach," *IEEE Transaction on Parallel and Distributed Systems*, vol. 10, pp. 44-61, 1999.

[22] S. M. Tang, Y. L. Wang and Y. H. Leu, "Optimal independent spanning trees on hypercubes," *Journal of Information Science and Engineering*, vol. 20, pp. 143-155, 2004.

[23] D. B. West, Introduction to Graph Theory, 2nd ed. Prentice Hall, Upper Saddle River, NJ, 2001.

[24] J. S. Yang, J. M. Chang, S. M. Tang and Y. L. Wang, "Reducing the height of inedpendent spanning trees in chordal rings," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, pp. 644-657, 2007.

[25] X. Yang, D. J. Evans, and G. M. Megson, "The locally twisted cubes," *International Journal of Computer Mathematics*, vol. 82, pp. 401-413, 2005.

[26] J. S. Yang, S. M. Tang, J. M. Chang and Y. L. Wang, "Parallel construction of optimal independent spanning trees on hypercubes," *Parallel Computing*, vol. 33, pp. 73-79, 2007.

[27] A. Zwhavi and A. Itai, "Three tree-paths," *Journal of Graph Theory*, vol. 13, pp. 175-188, 1989.