วิธีการแบบทั่วไปในการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีแบบขนานบนระบบเครือข่ายแบบไฮเปอร์คิว

GENERALIZED PARALLEL CONSTRUCTION OF OPTIMAL INDEPENDENT SPANNING TREES ON HYPERCUBES

สุมิตรา ขจรเงิน
SUMITRA KHAJONNGOEN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2552

KMITL-2009-SC-M-002-019

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

วิธีการแบบทั่วไปในการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีแบบขนานบนระบบเครือข่ายแบบไฮเปอร์คิว

GENERALIZED PARALLEL CONSTRUCTION OF OPTIMAL INDEPENDENT SPANNING TREES ON HYPERCUBES

สุมิตรา ขจรเงิน

SUMITRA KHAJONNGOEN

เลขหมู่.....
เลขทะเบียน.105312
วัน,เดือน,ปี.....18 พ.ย. 2552

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL – 2009 – SC – M – 002 - 019

GENERALIZED PARALLEL CONSTRUCTION OF OPTIMAL INDEPENDENT SPANNING TREES ON HYPERCUBES

SUMITRA KHAJONNGOEN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2009

KMITL – 2009 – SC – M – 002 – 019

COPYRIGHT 2009

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ วิธีการแบบทั่วไปในการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีแบบขนานบนระบบเครือข่ายแบบไฮเปอร์คิว

Generalized Parallel Construction of Optimal Independent Spanning Trees on Hypercubes

นักศึกษา นางสาวสุมิตรา ขจรเงิน

รหัสประจำตัว 50067507

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.ดร.จีรพร วีระพันธุ์

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.วีระ	บุญจริง	
ผศ.ดร.จีรพร	วีระพันธุ์	
ผศ.ดร.ศรัณย์	อินทโกสุม	
ดร.เฉลิมศักดิ์	เลิศวงศ์เสถียร	

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

วัน / เดือน/ ปี ที่สอบ 18 พฤษภาคม 2552 เวลา 14.00 น. เป็นต้นไป

สถานที่สอบ ณ อาคารจุฬาภรณวลัยลักษณ์ 1 ห้อง 217

คณะวิทยาศาสตร์รับรองแล้ว

(รองศาสตราจารย์ ดร.ธีรวัฒน์ มงคลอัศวรัตน์)

คณบดีคณะวิทยาศาสตร์

วันที่ 25 **เดือน** พฤษภาคม **พ.ศ.** 2552

หัวข้อวิทยานิพนธ์	วิธีการแบบทั่วไปในการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีแบบขนานบนระบบเครือข่ายแบบไฮเปอร์คิว
นักศึกษา	นางสาวสุมิตรา ขจรเงิน
รหัสประจำตัว	50067507
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.จีรพร วีระพันธุ์

บทคัดย่อ

คุณประโยชน์อย่างหนึ่งของต้นไม้ทอดข้ามที่แตกต่างในการกระจายข้อมูลไปบนระบบเครือข่ายแบบขนานคือการช่วยเพิ่มความน่าเชื่อถือของระบบเครือข่าย เมื่อไม่นานมานี้มีผลงานวิจัยที่นำเสนอขั้นตอนวิธีแบบขนานเพื่อสร้างต้นไม้ทอดข้ามที่แตกต่างที่เหมาะสมที่สุด k ต้นโดยมีโหนดราก (โหนดต้นทางในการกระจายข้อมูล) เป็นโหนด 0 บนระบบเครือข่ายแบบไฮเปอร์คิว k มิติ ด้วยความซับซ้อนด้านเวลาเท่ากับ O(k) ซึ่งต้นไม้ที่ได้จากวิธีการดังกล่าวไม่สามารถทำการกระจายข้อมูลจากโหนดอื่นที่ไม่ใช่โหนด 0 ได้ จากเหตุผลดังกล่าวงานวิจัยนี้จึงนำเสนอขั้นตอนวิธีทั่วไปแบบขนานเพื่อสร้างต้นไม้ทอดข้ามที่แตกต่างที่เหมาะสมที่สุด k ต้นที่มีโหนดรากเป็นโหนดใดๆ (0, 1, 2, …, หรือ 2^k-1) บนระบบเครือข่ายแบบไฮเปอร์คิว k มิติ โดยใช้หลักการที่เรียกว่าจัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไป (Generalized Hamming Distance Latin Square : GHDLS) ด้วยความซับซ้อนด้านเวลาเท่ากับ O(k)

Thesis	Generalized Parallel Construction of Optimal Independent Spanning Trees on Hypercubes
Student	Miss Sumitra Khajonngoen
Student ID.	50067507
Degree	Master of Science
Program	Computer Science
Year	2009
Thesis Advisor	Asst. Prof. Dr. Jeeraporn Werapun

ABSTRACT

An advantage of independent spanning trees (ISTs) for data broadcasting on any parallel network is the increase of network reliability. Recently work proposed a parallel algorithm for constructing the k optimal ISTs rooted at vertex 0 (as a source node in broadcasting) on k-dimensional (k-D) hypercubes in O(k) time. In this paper, we proposed a generalized parallel algorithm that constructs the k optimal ISTs on k-D hypercubes, based on the Generalized Hamming Distance Latin Square (GHDLS) scheme. Our new parallel algorithm can construct k ISTs rooted at an arbitrary vertex (0, 1, 2, …, or 2^k-1) in O(k) time and minimize the height of ISTs.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้มิอาจจะสำเร็จลุล่วงไปได้ด้วยดี หากมิได้รับคำแนะนำ คำชี้แจง ความรู้และความเอาใจใส่จาก ผศ.ดร.จีรพร วีระพันธุ์ ผู้เป็นอาจารย์ที่ปรึกษา ซึ่งท่านได้สละเวลาให้กับข้าพเจ้าอย่างเต็มที่ จึงใคร่ขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ รศ.ดร.วีระ บุญจริง ผศ.ดร.ศรัณย์ อินทโกสุม และดร.เฉลิมศักดิ์ เลิศวงศ์เสถียร คณะกรรมการสอบวิทยานิพนธ์ ที่กรุณาให้คำแนะนำตลอดจนข้อชี้แนะจนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบพระคุณมหาวิทยาลัยศรีนครินทรวิโรฒ ที่สนันสนุนและให้โอกาสในการศึกษาต่อในระดับที่ตั้งใจ

ขอขอบพระคุณทุนสนับสนุนการทำวิทยานิพนธ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้การสนับสนุนทุนการศึกษาแกข้าพเจ้า

ขอขอบพระคุณบิดา มารดา ที่สนันสนุนและเป็นกำลังใจในระหว่างการศึกษาเป็นอย่างดียิ่ง

ขอขอบคุณ นางสาวพัชรินทร์ บัวเย็น นางสาวรสลิน เพตะกร นายขวัญชัย เล้าสุขสุวรรณ พี่ๆ และเพื่อนๆ ทุกคนที่ให้คำปรึกษา และช่วยอำนวยความสะดวกในด้านต่างๆ

สำหรับคุณงามความดีและประโยชน์อันใดที่เกิดขึ้นจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดา มารดา อาจารย์ทุกท่านซึ่งเป็นที่เคารพรักยิ่ง ตลอดจนญาติพี่น้อง และเพื่อนๆ ทุกคน

สุมิตรา ขจรเงิน

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่ออังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 2

1.3 สมมติฐานของการศึกษา 3

1.4 ขอบเขตการวิจัย 3

1.5 ขั้นตอนการศึกษาและการดำเนินงานวิจัย 3

1.6 ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 เครือข่ายแบบไฮเปอร์คิว 5

2.2 การกระจายข้อมูลบนระบบเครือข่ายแบบไฮเปอร์คิว 7

2.3 ต้นไม้ทอดข้ามที่แตกต่าง 8

2.4 จัตุรัสลาติน 9

2.5 การสร้างต้นไม้ทอดข้ามที่แตกต่างบนเครือข่ายแบบไฮเปอร์คิว 9

2.5.1 ความรู้เบื้องต้นที่เกี่ยวข้องกับการสร้างต้นไม้ทอดข้ามที่แตกต่าง 10

2.5.1.1 ความหมายของสัญลักษณ์ที่ใช้ในขั้นตอนวิธี 10

2.5.1.2 คุณสมบัติที่ควรทราบสำหรับขั้นตอนวิธี 10

2.5.2 ขั้นตอนวิธีสำหรับสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนด 0 บนเครือข่ายแบบไฮเปอร์คิว 13

2.6 การพิสูจน์ความถูกต้องของขั้นตอนวิธี 17

สารบัญ (ต่อ)

หน้า

บทที่ 3 การสร้างต้นไม้ทอดข้ามที่แตกต่างแบบใหม่บนระบบเครือข่ายแบบไฮเปอร์คิว 24

3.1 ความหมายของสัญลักษณ์ต่างๆ ที่ใช้ในขั้นตอนวิธี 24

3.2 ความรู้พื้นฐานที่จำเป็นสำหรับขั้นตอนวิธี 25

3.3 ขั้นตอนวิธีสำหรับสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนดใดๆ บนเครือข่ายแบบไฮเปอร์คิว 27

บทที่ 4 การวิเคราะห์ขั้นตอนวิธี 57

4.1 การวิเคราะห์ความซับซ้อนด้านเวลา 57

4.2 การวิเคราะห์ความถูกต้อง 58

บทที่ 5 บทสรุป 66

5.1 บทสรุป 66

5.2 แนวทางการพัฒนางานวิจัย 67

เอกสารอ้างอิง 68

ภาคผนวก 70

ผลงานวิจัยที่ได้รับการตีพิมพ์

ประวัติผู้เขียน 77

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_4 (i = 0) 14

2.2 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_4 (i = 1) 15

2.3 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_4 (i = 2) 16

2.4 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_3 บน Q_4 (i = 3) 17

2.5 แสดงตัวอย่างการคำนวณหาโหนดลูกของโหนด 0 ตั้งแต่ T_0 ถึง T_3 ของ Q_4 18

3.1 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 0, i = 0) 29

3.2 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 0, i = 1) 30

3.3 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 0, i = 2) 30

3.4 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 1, i = 0) 31

3.5 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 1, i = 1) 32

3.6 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 1, i = 2) 33

3.7 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 2, i = 0) 34

3.8 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 2, i = 1) 34

3.9 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 2, i = 2) 35

3.10 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 3, i = 0) 36

3.11 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 3, i = 1) 37

3.12 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 3, i = 2) 37

3.13 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 4, i = 0) 38

3.14 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 4, i = 1) 39

3.15 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 4, i = 2) 40

3.16 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 5, i = 0) 41

3.17 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 5, i = 1) 41

3.18 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 5, i = 2) 42

3.19 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 6, i = 0) 43

3.20 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 6, i = 1) 44

3.21 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 6, i = 2) 44

3.22 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 7, i = 0) 45

3.23 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 7, i = 1) 46

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

3.24 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 7, i = 2).....47

3.25 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_4 (r = 8, i = 0).....48

3.26 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_4 (r = 8, i = 1).....49

3.27 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_4 (r = 8, i = 2).....50

3.28 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_3 บน Q_4 (r = 8, i = 3).....51

3.29 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_4 (r = 13, i = 0).....52

3.30 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_4 (r = 13, i = 1).....53

3.31 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_4 (r = 13, i = 2).....54

3.32 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_3 บน Q_4 (r = 13, i = 3).....55

4.1 แสดงตัวอย่างการคำนวณหาโหนดลูกของโหนด 8(1000) ตั้งแต่ T_0 ถึง T_3 ของ Q_4.....59

สารบัญรูป

รูปที่ หน้า

2.1 เครือข่ายแบบไฮเปอร์คิว 1-มิติ 2-มิติ และ 3-มิติ 5

2.2 ขั้นตอนการสร้างไฮเปอร์คิว 3 มิติ ($Q_3 = Q_2 \times Q_1$) 6

2.3 ระยะทางระหว่างโหนด 2 และ 5 บนไฮเปอร์คิว 3 มิติ โดยผ่านโหนด 3 และโหนด 1 ตามลำดับ 7

2.4 ตัวอย่างการกระจายข้อมูลจากโหนด 0 บนไฮเปอร์คิว 3 มิติ 8

2.5 ตัวอย่างต้นไม้ทอดข้ามที่แตกต่างบน Q_3 ที่มีโหนด 0 เป็นโหนดราก 9

2.6 แสดงโหนดประชิดของโหนด 0 บน Q_4 10

2.7 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ที่มีโหนดรากเป็นโหนด 0 11

2.8 แสดงเส้นทางที่ยาวที่สุดใน T_3 ที่มีโหนดรากเป็นโหนด 0 บน Q_4 19

3.1 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0,T_1,T_2 และ T_3 ของ Q_4 ที่มีโหนดรากเป็นโหนด 8 (r=8) 25

3.2 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 0 (r = 0) 28

3.3 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 1 (r = 1) 31

3.4 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 2 (r = 2) 33

3.5 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 3 (r = 3) 35

3.6 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 4 (r = 4) 38

3.7 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 5 (r = 5) 40

3.8 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 6 (r = 6) 42

3.9 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 7 (r = 7) 45

3.10 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ที่มีโหนดรากเป็นโหนด 13 (r=13) 51

4.1 แสดงเส้นทางที่ยาวที่สุดใน T_3 ที่มีโหนดรากเป็นโหนด 8(1000) บน Q_4 60

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

กราฟถือได้ว่ามีบทบาทสำคัญต่อระบบการประมวลผลแบบขนาน (Parallel computing system) โดยใช้อธิบายองค์ประกอบของระบบเครือข่าย (Network system) ซึ่งจะช่วยให้เห็นภาพรวมของระบบเครือข่ายและช่วยให้การแก้ปัญหาของระบบเครือข่ายทำได้ง่ายยิ่งขึ้น หากพิจารณากราฟ G(V,E) ใดๆ โหนดต่างๆ (V(G)) ใช้แทนหน่วยประมวลผล (Processor) ในระบบ และใช้เส้นเชื่อม (E(G)) เพื่อแทนลิงค์ต่างๆ ที่เชื่อมระหว่างหน่วยประมวลผล

ปัญหาบนระบบเครือข่ายปัญหาหนึ่งที่สามารถประยุกต์ใช้กราฟเพื่อแก้ปัญหาได้อย่างมีประสิทธิภาพคือ ปัญหาการกระจายข้อมูล (Data broadcasting) เมื่อพิจารณาระบบเครือข่ายในรูปแบบของกราฟ การกระจายข้อมูลคือการส่งข้อมูลจากโหนดต้นทางไปยังทุกๆ โหนดที่อยู่ในระบบ (One-to-all) ซึ่งปัญหาที่เกิดขึ้นในกระบวนการนี้คือ การส่งข้อมูลไม่สำเร็จหรือการแพร่ของข้อมูลไม่สามารถไปถึงโหนดปลายทางบางโหนดได้ ในกรณีที่ทุกๆ โหนดในระบบสามารถทำการรับส่งข้อมูลได้ปกติ สาเหตุของปัญหาอาจเกิดจากลิงค์ระหว่างโหนดคู่ใดๆ ขัดข้องหรือมีการรับส่งข้อมูลผ่านลิงค์นั้นเป็นจำนวนมากจนกระทั่งลิงค์ไม่สามารถรับหรือส่งข้อมูลในขณะนั้นได้ วิธีหนึ่งที่ใช้ในการแก้ปัญหาดังกล่าวทำได้โดยการสำเนาข้อมูลและส่งผ่านไปยังเส้นทางสำรองที่ไม่ผ่านลิงค์ที่มีปัญหา และเพื่อแก้ปัญหาในกรณีที่มีลิงค์ที่ขัดข้องมากกว่าหนึ่งลิงค์ เส้นทางดังกล่าวควรจะมีหลายชุดและต้องเป็นเส้นทางที่แตกต่างกัน ซึ่งการกระจายข้อมูลในลักษณะดังกล่าวมีความน่าเชื่อถือหรือทนทานต่อความผิดพลาดสูง (Fault-Tolerant Broadcasting) [9]

จากปัญหาดังกล่าวได้มีการศึกษาการหาชุดเส้นทางที่แตกต่างกันบนกราฟเครือข่าย G(V,E) โดยพิจารณาเส้นทางการส่งข้อมูลในรูปแบบของโครงสร้างต้นไม้ที่เรียกว่า ต้นไม้ทอดข้ามที่แตกต่าง (Independent Spanning Trees : ISTs) ที่มีโหนดราก (root : r) เป็นโหนดต้นทางในการส่งข้อมูล ซึ่งการกระจายข้อมูลที่ทนทานต่อความผิดพลาดสูงทำได้โดยสร้างต้นไม้ทอดข้ามที่แตกต่างจำนวน k ต้น ($k > 1$) และทำการแพร่ข้อมูลสำเนาจำนวน k ชุด ไปบนต้นไม้ทอดข้ามที่แตกต่าง k ต้นนั้น หากโหนดรากหรือโหนดต้นทางไม่เกิดความผิดพลาดใดๆ การกระจายข้อมูลในลักษณะนี้จะสามารถทนทานต่อความผิดพลาดได้ถึง k เท่า

ปัญหาการสร้างต้นไม้ทอดข้ามที่แตกต่างหลายชุดได้รับความสนใจและมีการศึกษาอย่างกว้างขวางบนเครือข่ายหลายรูปแบบด้วยกัน เช่น เครือข่ายบนกราฟเชิงระนาบ (Planar graphs network) [3, 4, 5], เครือข่ายแบบกราฟผลคูณ (Product graphs network) [7], เครือข่ายแบบคอร์ดอล

ริง (Chordal rings network) [6], เครือข่ายทอรัสแบบหลายมิติ (Multidimensional tori network) [12], ระบบเครือข่ายแบบไดนามิก (Dynamic network) [8] และเครือข่ายแบบไฮเปอร์คิว (Hypercubes network) [11, 13] โดยจุดมุ่งหมายของงานวิจัยข้างต้นมี 2 แนวทางคือ 1. เพื่อพัฒนาขั้นตอนวิธีในการสร้างต้นไม้ทอดข้ามที่แตกต่างให้มีประสิทธิภาพในแง่ของความซับซ้อนด้านเวลา 2. พัฒนาขั้นตอนวิธีเพื่อลดความสูงของต้นไม้ทอดข้ามที่แตกต่าง เมื่อไม่นานมานี้ได้มีงานวิจัยที่ศึกษาเกี่ยวกับการสร้างต้นไม้ทอดข้ามที่แตกต่างแบบขนานบนเครือข่ายแบบไฮเปอร์คิว k มิติ (Parallel construction of optimal ISTs on k-dimensional hypercubes) ซึ่งมีโหนดรากเป็นโหนด 0 โดยใช้หลักการที่เรียกว่า จัตุรัสลาตินของระยะทางแฮมมิ่ง (Hamming Distance Latin Square Scheme: HDLS) จากหลักการนี้ทำให้สามารถสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิว k มิติ ได้ด้วยความซับซ้อนด้านเวลาเท่ากับ O(k) เนื่องจากเครือข่ายแบบไฮเปอร์คิวมีโครงสร้างแบบโหนดสมมาตร (Vertex-symmetric) ซึ่งทำให้สามารถกำหนดให้โหนดใดๆ ก็ได้บนเครือข่ายเป็นโหนด 0 แต่อย่างไรก็ตามโหนดแต่ละโหนดบนระบบเครือข่ายจะถูกกำหนดหมายเลขประจำโหนดเพียงหนึ่งครั้งในช่วงแรก (Initial setting) และจะไม่สามารถปรับเปลี่ยนหมายเลขได้อีกในขณะที่มีการประมวลผลอยู่ (Runtime) ภายใต้หลักการนี้จะเห็นได้ว่าในการกระจายข้อมูลจากโหนด 0 สามารถทำได้ แต่หลังจากมีการจัดสรรหมายเลขประจำโหนด (Allocation time) แล้วจะไม่สามารถทำการกระจายข้อมูลจากโหนดอื่นที่ไม่ใช่โหนด 0 ได้เลย

จากเหตุผลข้างต้นงานวิจัยนี้จึงได้นำเสนอวิธีการแบบทั่วไปในการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีแบบขนานบนระบบเครือข่ายแบบไฮเปอร์คิว k มิติ (Generalized parallel construction of optimal ISTs on k-dimensional hypercubes) ซึ่งสามารถสร้างต้นไม้ทอดข้ามที่แตกต่างจำนวน k ต้นที่มีโหนดรากเป็นโหนดใดๆ (0, 1, 2, ..., หรือ 2^k-1) บนระบบเครือข่ายโดยขั้นตอนวิธีนี้เป็นรูปแบบทั่วไปของขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างแบบขนานบนเครือข่ายแบบไฮเปอร์คิว k มิติ ซึ่งมีโหนดรากเป็นโหนด 0 ที่มีความซับซ้อนด้านเวลาเท่ากับ O(k)

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อศึกษาวิธีการแบบทั่วไปในการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีแบบขนานบนเครือข่ายแบบไฮเปอร์คิว ดังนี้

ออกแบบและปรับปรุงขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนด 0 ให้สามารถสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนดใดๆ (0, 1, 2, ..., หรือ 2^k-1)

1.3 สมมติฐานของการศึกษา

การเพิ่มประสิทธิภาพการกระจายข้อมูลบนเครือข่ายแบบไฮเปอร์คิว k มิติให้สามารถกระจายข้อมูลจากโหนดใดๆ (0, 1, 2, ..., หรือ 2^k-1) โดยการออกแบบและปรับปรุงขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง k ต้น ด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนด 0 เพื่อให้สามารถสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนดใดๆ ด้วยความซับซ้อนด้านเวลาเท่ากับ O(k)

1.4 ขอบเขตการวิจัย

วิทยานิพนธ์นี้มีขอบเขตการวิจัยเพื่อทำการศึกษาขั้นตอนการสร้างต้นไม้ทอดข้ามที่แตกต่าง k ต้น ด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนดใดๆ (0, 1, 2, ..., หรือ 2^k-1)

1.5 ขั้นตอนการศึกษาและดำเนินงานวิจัย

วิทยานิพนธ์นี้มีขั้นตอนการศึกษาและการดำเนินงานวิจัย ดังนี้

1) ศึกษางานวิจัยที่เกี่ยวข้องกับเครือข่ายแบบไฮเปอร์คิวและขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างบนเครือข่ายแบบไฮเปอร์คิว (Hypercubes) ทั้งแบบขนาน (Parallel) และแบบต่อเนื่อง (Sequential)

2) ทำการตั้งสมมติฐานโดยคาดว่าการปรับปรุงหลักการของจัตุรัสลาตินของระยะทางแฮมมิ่ง (Hamming Distance Latin Square Scheme: HDLS) ให้อยู่ในรูปแบบทั่วไปจะสามารถสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนดใดๆ ได้

3) ออกแบบขั้นตอนวิธีใหม่ จากการปรับปรุงวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนด 0 เพื่อใช้สร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนดใดๆ

4) พิสูจน์ความถูกต้อง (Proof of correctness) ของขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนดใดๆ สรุปผล พร้อมเสนอแนวทางการพัฒนางานวิจัย

5) เขียนวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

วิทยานิพนธ์นี้มีประโยชน์ที่คาดว่าจะได้รับ ดังนี้

1) ทำให้ได้ขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวที่มีโหนดรากเป็นโหนดใดๆ

2) สามารถนำแนวคิดจากขั้นตอนวิธีที่นำเสนอ ไปประยุกต์ใช้เพื่อเพิ่มประสิทธิภาพการกระจายข้อมูลบนเครือข่ายแบบไฮเปอร์คิว เพื่อให้มีความน่าเชื่อถือ (Reliability) และมีความทนทานต่อความผิดพลาดของระบบเครือข่าย (Fault-Tolerant Broadcasting)

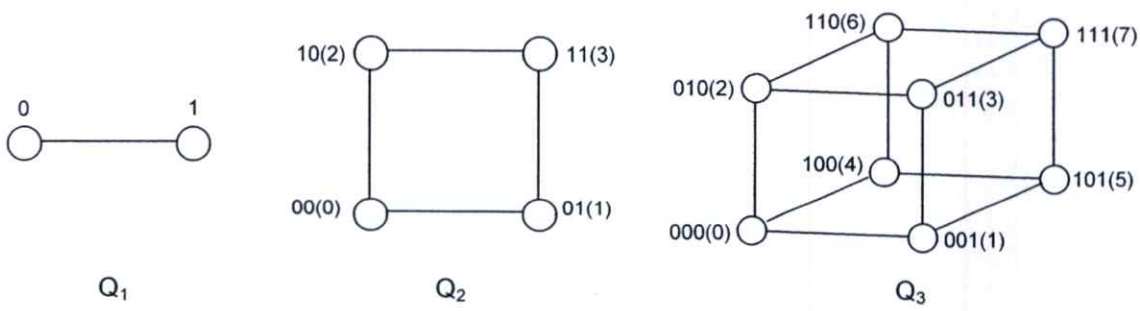
บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงคุณลักษณะทั่วไปและรูปแบบการกระจายข้อมูลของเครือข่ายแบบไฮเปอร์คิว ทฤษฎีที่เกี่ยวข้องและขั้นตอนวิธีแบบขนานในการสร้างต้นไม้ทอดข้ามที่แตกต่างบนเครือข่ายแบบไฮเปอร์คิว และในส่วนสุดท้ายจะเป็นการพิสูจน์ความถูกต้องของขั้นตอนวิธีดังกล่าว

2.1 เครือข่ายแบบไฮเปอร์คิว

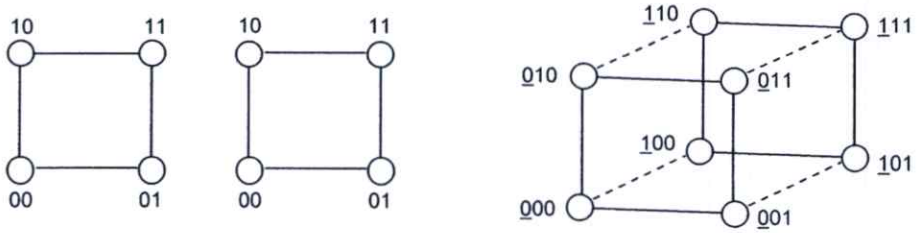
เครือข่ายแบบไฮเปอร์คิว (Hypercube network) เป็นระบบเครือข่ายแบบขนานชนิดหนึ่งที่สามารถนำไปประยุกต์ใช้กับการประมวลผลแบบขนานได้หลากหลายประเภท เนื่องจากระบบเครือข่ายดังกล่าวมีคุณสมบัติที่น่าสนใจหลายอย่าง เช่น โครงสร้างระบบสามารถสร้างได้ง่าย เส้นผ่าศูนย์กลางเครือข่ายเป็นแบบลอการิทึม (Logarithmic diameter) อีกทั้งมีความเหมาะสมในการนำไปใช้สำหรับพัฒนาขั้นตอนวิธีแบบขนาน และคุณสมบัติที่สำคัญอย่างมากประการหนึ่งคือความทนทานต่อความผิดพลาดสูง (Fault-tolerance)

ไฮเปอร์คิว k มิติ (k-dimensional hypercube) หรือ Q_k คือ ระบบเครือข่ายที่มีหน่วยประมวลผลหลายหน่วย ($N = 2^k$) เชื่อมต่อกันด้วยเส้นเชื่อม k เส้นต่อหน่วยประมวลผล ตัวอย่างของไฮเปอร์คิว 1-มิติ (Q_1) 2-มิติ (Q_2) และ 3-มิติ (Q_3) แสดงดังรูปที่ 2.1 ซึ่ง Q_k สามารถสร้างได้จากผลคูณคาร์ทิเชียน (Cartesian product) ระหว่าง Q_{k-1} และ Q_1 ($Q_k = Q_{k-1} \times Q_1$ โดยที่ x แทนการคูณแบบคาร์ทิเชียน) [13] ขั้นตอนการสร้างไฮเปอร์คิว 3 มิติ ซึ่งเกิดจากผลคูณคาร์ทิเชียนระหว่าง Q_2 และ Q_1 แสดงดังรูปที่ 2.2



รูปที่ 2.1 เครือข่ายแบบไฮเปอร์คิว 1-มิติ 2-มิติ และ 3-มิติ

หากพิจารณาระบบในรูปแบบของกราฟ กราฟดังกล่าวจะประกอบด้วย $N = 2^k$ โหนด เมื่อ N คือจำนวนโหนดทั้งหมดของไฮเปอร์คิว โดยมีจำนวนลิงค์ระหว่างโหนดต่างๆ ทั้งหมดเท่ากับ kN/2 มีดีกรีของโหนดเท่ากับ k และมีเส้นผ่าศูนย์กลางเครือข่ายเท่ากับ $\log_2 N$ จากคุณสมบัตินี้จะเห็นได้ว่าขนาดของเส้นผ่าศูนย์กลางเครือข่ายของไฮเปอร์คิวจะเติบโตช้าเมื่อเทียบกับจำนวนโหนดที่เพิ่มขึ้น ซึ่งถือได้ว่าเป็นคุณลักษณะที่ดีอย่างหนึ่งของระบบเครือข่ายนี้ ในเครือข่ายแบบไฮเปอร์คิวหมายเลขประจำโหนดของแต่ละโหนดจะถูกแทนด้วยเลขฐานสองจำนวน k บิต ตั้งแต่บิตที่ 0 ถึงบิตที่ k-1 ($b_{k-1}b_{k-2}...b_1b_0$) และมีหมายเลขประจำโหนดตั้งแต่ 0 ถึง 2^k-1 ยกตัวอย่างเช่น ไฮเปอร์คิว 3 มิติ (k = 3) จะมีหมายเลขประจำโหนดในรูปแบบของเลขฐานสองทั้งหมด 3 บิต ตั้งแต่บิตที่ 0 ถึงบิตที่ 2 ดังนี้ $b_2b_1b_0$ โดยที่ b_i แทนค่าของบิตในตำแหน่งที่ i (i = 0, 1, 2) บนหมายเลขประจำโหนด และมีหมายเลขประจำโหนด = $\sum_{i=0}^{k} b_i 2^i$ ซึ่งเป็นค่าในรูปแบบของเลขฐานสิบตั้งแต่ 0 ถึง 7 โดยแสดงในรูปแบบของเลขฐานสองดังนี้ 000(0), 001(1), 010(2), 011(3), 100(4), 101(5), 110(6), 111(7)



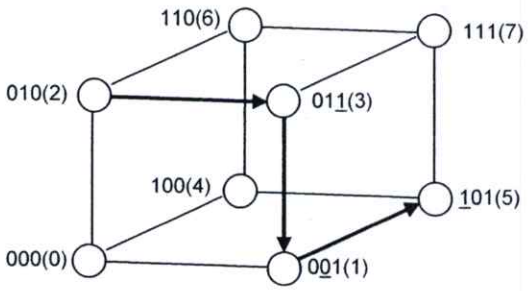
(ก) ขั้นที่ 1 ทำสำเนา Q_2 ไว้เท่ากับจำนวนสมาชิกของ Q_1

(ข) ขั้นที่ 2 สร้างเส้นเชื่อมระหว่าง Q_2 ทั้ง 2 เครือข่าย (-----) ทำให้ได้เครือข่ายใหม่เป็น Q_3

รูปที่ 2.2 ขั้นตอนการสร้างไฮเปอร์คิว 3 มิติ ($Q_3 = Q_2 \times Q_1$)

คุณสมบัติที่สำคัญอีกประการหนึ่งของไฮเปอร์คิว k มิติคือ เลขฐานสองของโหนดที่ติดกันจะต่างกันเพียงหนึ่งบิตเท่านั้น จากคุณสมบัตินี้หากพิจารณาระยะทางระหว่างโหนดสองโหนดบนเครือข่ายโดยการนับจำนวนเส้นเชื่อมทั้งหมดบนเส้นทาง จะพบว่าระยะทางระหว่างโหนด A และโหนด B สามารถหาได้จากการนับบิตที่ต่างกันของทั้งสองโหนดหรือระยะทางแฮมมิ่ง (Hamming distance) ระหว่าง A และ B [10] ยกตัวอย่างบนไฮเปอร์คิว 3 มิติ สมมติให้ A เป็นโหนด 010(2) และโหนด B เป็นโหนด 101(5) ซึ่งมีระยะทางแฮมมิ่งเท่ากับ 3 ดังนั้นจะได้ว่า

ระยะทางระหว่างโหนด 2 และ 5 บนไฮเปอร์คิว 3 มิติเท่ากับ 3 เส้นเชื่อมคือ 2(010) → 3(011) → 1(001) → 5(101) ตัวอย่างแสดงดังรูป 2.3



รูปที่ 2.3 ระยะทางระหว่างโหนด 2 และ 5 บนไฮเปอร์คิว 3 มิติ โดยผ่านโหนด 3 และโหนด 1 ตามลำดับ

2.2 การกระจายข้อมูลบนเครือข่ายแบบไฮเปอร์คิว

การกระจายข้อมูลจากหน่วยประมวลผลหนึ่งไปยังหน่วยประมวลอื่นทั้งหมดบนเครือข่ายแบบไฮเปอร์คิว ทำได้โดยการส่งข้อมูลไปยังโหนดที่มีเส้นเชื่อมต่อกันในแต่ละมิติของไฮเปอร์คิว โดยจะเริ่มกระจายข้อมูลจากมิติต่ำสุด (Lowest dimension) ซึ่งก็คือมิติที่พิจารณาจากบิตต่ำสุด (Least significant bit) ของเลขฐานสองของหมายเลขประจำโหนด หรือเริ่มกระจายข้อมูลจากมิติสูงสุด (Highest dimension) หรือมิติที่พิจารณาจากบิตสูงสุด (Most significant bit) ของเลขฐานสองก็ได้ [2] ซึ่งระยะเวลาที่ใช้ในการกระจายข้อมูลบนระบบเครือข่ายนี้เท่ากับ $\log_2 N$ เมื่อ N คือจำนวนโหนดทั้งหมดบนระบบเครือข่าย ยกตัวอย่างเช่น การกระจายข้อมูลจากโหนด 0 ไปยังโหนดอื่นๆ บนไฮเปอร์คิว 3 มิติ (N=8, $\log_2 N = 3$) โดยเริ่มกระจายข้อมูลจากมิติต่ำสุดซึ่งทำได้ใน 3 ขั้นตอนดังนี้

ขั้นที่ 1 (พิจารณาบิตต่ำสุด)	โหนด 000(0) ส่งข้อมูลไปยังโหนด 001(1)
ขั้นที่ 2 (พิจารณาบิตต่ำรองลงมา)	โหนด 000(0) ส่งข้อมูลไปยังโหนด 010(2)
	โหนด 001(1) ส่งข้อมูลไปยังโหนด 011(3)
ขั้นที่ 3 (พิจารณาบิตสูงสุด)	โหนด 000(0) ส่งข้อมูลไปยังโหนด 100(4)
	โหนด 001(1) ส่งข้อมูลไปยังโหนด 101(5)
	โหนด 010(2) ส่งข้อมูลไปยังโหนด 110(6)
	โหนด 011(3) ส่งข้อมูลไปยังโหนด 111(7)

ส่วนการกระจายข้อมูลจากโหนด 0 โดยเริ่มกระจายข้อมูลจากมิติสูงสุดทำได้ใน 3 ขั้นตอนดังนี้

ขั้นที่ 1 (พิจารณาบิตสูงสุด) โหนด 000(0) ส่งข้อมูลไปยังโหนด 100(4)

ขั้นที่ 2 (พิจารณาบิตสูงรองลงมา) โหนด 000(0) ส่งข้อมูลไปยังโหนด 010(2)

โหนด 100(4) ส่งข้อมูลไปยังโหนด 110(6)

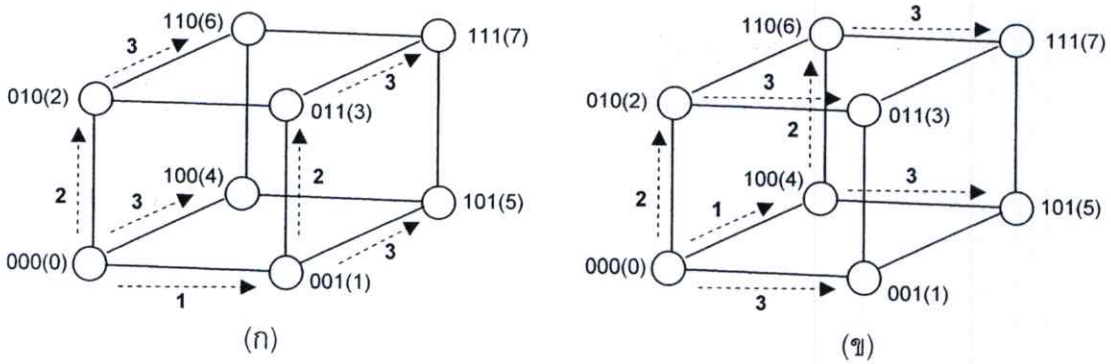
ขั้นที่ 3 (พิจารณาบิตต่ำสุด) โหนด 000(0) ส่งข้อมูลไปยังโหนด 001(1)

โหนด 010(2) ส่งข้อมูลไปยังโหนด 011(3)

โหนด 100(4) ส่งข้อมูลไปยังโหนด 101(5)

โหนด 110(6) ส่งข้อมูลไปยังโหนด 111(7)

ตัวอย่างการกระจายข้อมูลทั้งแบบเริ่มกระจายจากมิติต่ำสุดและมิติสูงสุด แสดงดังรูปที่ 2.4 (ก) และ 2.4 (ข) ตามลำดับ



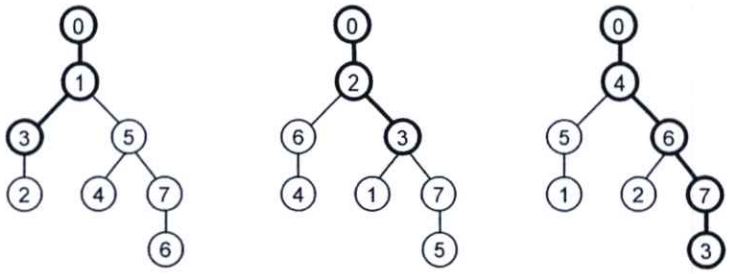
รูปที่ 2.4 ตัวอย่างการกระจายข้อมูลจากโหนด 0 บนไฮเปอร์คิว 3 มิติ

2.3 ต้นไม้ทอดข้ามที่แตกต่าง (Independent spanning tree)

J.-S. Yang และคณะ [13] ได้อธิบายความหมายของต้นไม้ทอดข้ามที่แตกต่างไว้ดังต่อไปนี้

นิยาม 2.1 เมื่อพิจารณาบนกราฟ G ใดๆ ซึ่งประกอบด้วยเซตของโหนด V(G) และเซตของเส้นเชื่อม E(G) ทั้งหมดบนกราฟ G และต้นไม้ T ใดๆ จะเรียกได้ว่าเป็นต้นไม้ทอดข้ามของ G ก็ต่อเมื่อ T ประกอบด้วยโหนดทุกโหนดและเส้นเชื่อมบางเส้นของ G และต้นไม้ทอดข้าม T และ T′ ใดๆ ของ G จะเรียกได้ว่าเป็น "ต้นไม้ทอดข้ามที่แตกต่าง" ได้ก็ต่อเมื่อ T และ T′ มีโหนดราก (r) เป็นโหนดเดียวกัน แต่เส้นทางทุกๆ เส้นทางจากโหนด $v \in V(G)$ โดยที่ $v(\neq r)$ ไปยัง r จะต้องแตกต่างกันในทุกลิงค์และทุกเส้นทาง

รูปที่ 2.5 แสดงตัวอย่างต้นไม้ทอดข้ามที่แตกต่าง 3 ต้น บน Q_3 ซึ่งมีโหนดรากเป็นโหนด 0 แต่เส้นทางจากโหนด 3 ถึงโหนด 0 ในทุกๆ ต้นจะแตกต่างกันในทุกๆ ลิงค์



รูปที่ 2.5 ตัวอย่างต้นไม้ทอดข้ามที่แตกต่างบน Q_3 ที่มีโหนด 0 เป็นโหนดราก

2.4 จัตุรัสลาติน (Latin square)

จัตุรัสลาติน [1] คือเมตริกซ์ขนาด nxn โดยที่สมาชิกทั้ง n ตัวนี้จะเรียงแตกต่างกันไปในทุกแถวและคอลัมน์ จากการเรียงสับเปลี่ยนเช่นนี้จึงทำให้สมาชิกแต่ละตัวจะปรากฏเพียงหนึ่งครั้งเท่านั้นในแต่ละแถวและแต่ละคอลัมน์ จัตุรัสลาตินที่รู้จักกันโดยทั่วไปเป็นจัตุรัสลาตินแบบที่มี 1,2,…,n เป็นสมาชิกแถวแรกและมีการเรียงลำดับสมาชิกเป็นแบบหมุนวน (Cyclic permutation) ซึ่งเรียกอีกอย่างหนึ่งว่า เมตริกซ์แบบหมุนวน (Circulant matrix) แสดงดังเมตริกซ์ต่อไปนี้

$$\begin{bmatrix} 1 & 2 & 3 & \cdots & n \\ 2 & 3 & 4 & \cdots & 1 \\ 3 & 4 & 5 & \cdots & 2 \\ \vdots & & & & \\ n & 1 & 2 & \cdots & n-1 \end{bmatrix}$$

ในกรณีที่ n = 4 จัตุรัสลาตินที่ได้จะมีขนาด 4 x 4 แสดงดังเมตริกซ์ด้านล่าง

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}$$

2.5 การสร้างต้นไม้ทอดข้ามที่แตกต่างบนเครือข่ายแบบไฮเปอร์คิว

การสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิว (Parallel construction of optimal independent spanning trees on hypercubes) [13] ได้ถูกเสนอขึ้นโดย J.-S. Yang และคณะ งานวิจัยดังกล่าวเป็นการออกแบบต้นไม้ทอดข้ามที่แตกต่างหลายต้นบนเครือข่ายแบบไฮเปอร์คิว ซึ่งต้นไม้ทอดข้ามที่แตกต่างนี้สามารถนำไปประยุกต์ใช้เพื่อ

ช่วยเพิ่มความทนทานต่อการผิดพลาด (Fault-tolerant) ของระบบได้ โดยการส่งข้อมูลจำนวน k สำเนาไปตามเส้นทางของต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากโหนดเดียวกัน k ต้น หากโหนดรากหรือโหนดต้นทางไม่เกิดความผิดพลาดใดๆ การกระจายข้อมูลในลักษณะนี้จะสามารถทนทานต่อความผิดพลาดได้ถึง k เท่า

2.5.1 ความรู้เบื้องต้นที่เกี่ยวข้องกับการสร้างต้นไม้ทอดข้ามที่แตกต่าง

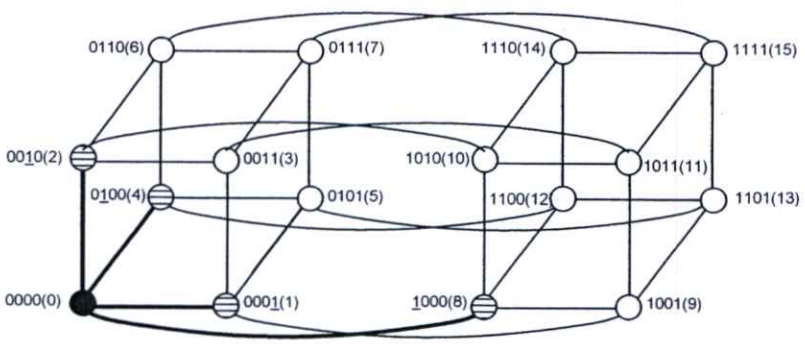
ในหัวข้อนี้จะเริ่มต้นด้วยการอธิบายความหมายของสัญลักษณ์ ต่อจากนั้นจะเป็นการแนะนำคุณสมบัติต่างๆ ที่ควรทราบ ซึ่งใช้ในขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างบนเครือข่ายแบบไฮเปอร์คิว

2.5.1.1 ความหมายของสัญลักษณ์ที่ใช้ในขั้นตอนวิธี

T_i	แทนต้นไม้ทอดข้ามที่แตกต่าง เมื่อ $0 \leq i \leq k-1$
T[x, r]	แทนเส้นทางจากโหนด x ถึงโหนดราก r บนต้นไม้ T
$N_G(x)$	แทนโหนดประชิด (Adjacent node) ของโหนด x ในกราฟ G
parent(T, x)	แทนโหนดพ่อแม่ของโหนด $x(\neq r)$ ซึ่งเป็นโหนดประชิดของ x บนเส้นทาง T[x, r]

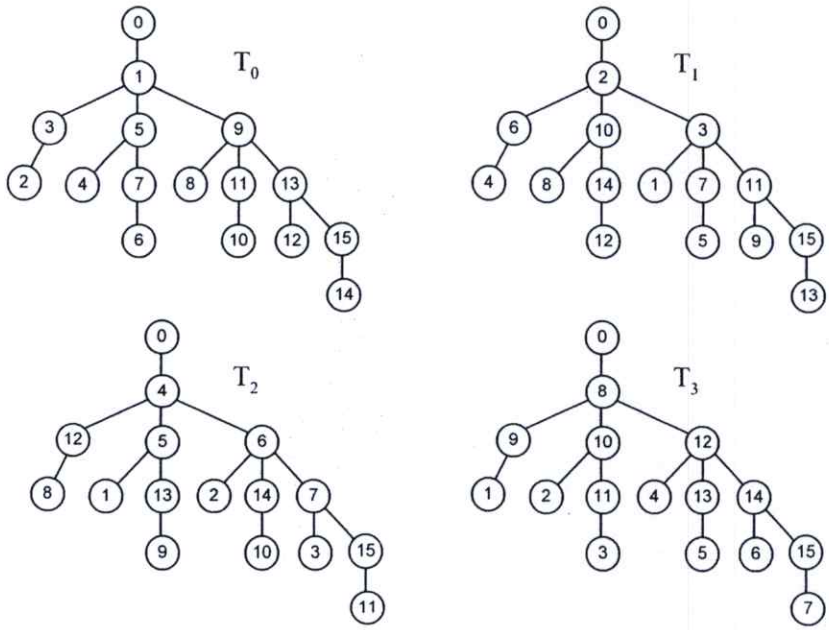
2.5.1.2 คุณสมบัติที่ควรทราบสำหรับขั้นตอนวิธี

เนื่องจากไฮเปอร์คิวเป็นเครือข่ายที่มีโหนดสมมาตร (Vertex-symmetric) ในที่นี้จึงกำหนดให้โหนด 0 เป็นโหนดรากของต้นไม้ทอดข้ามที่แตกต่างบนไฮเปอร์คิว k มิติ (Q_k) สมมติให้ τ เป็นเซตของต้นไม้ทอดข้ามที่แตกต่าง k ต้นที่มีโหนดรากเป็นโหนด 0 บน Q_k ($\tau = \{T_i: 0 \leq i \leq k-1\}$) จะเห็นได้ว่ามีโหนดประชิดของ 0 หรือ $N_{Q_k}(0) = \{2^0, 2^1, ..., 2^{k-1}\}$ ตัวอย่างโหนดประชิดของโหนด 0 บน Q_4 แสดงดังรูปที่ 2.6



รูปที่ 2.6 แสดงโหนดประชิดของโหนด 0 บน Q_4

จากคุณสมบัติการเชื่อมต่อของ Q_k และนิยามของ τ ทำให้โหนด 0 ของต้นไม้ทอดข้าม $T_i \in \tau$ มีโหนดลูกเพียงโหนดเดียวเท่านั้น หากกำหนดให้โหนด 2^i $(0 \leq i \leq k\text{-}1)$ เป็นโหนดลูกของโหนด 0 บน T_i แล้ว T_0, T_1, T_2 และ T_3 ของ Q_4 แสดงดังรูปที่ 2.7



รูปที่ 2.7 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ที่มีโหนดรากเป็นโหนด 0

สำหรับ Q_k ใดๆ ซึ่งมีโหนด $x(\neq 0)$ เป็นสมาชิก โดยหมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ $x = x_{k-1}x_{k-2}\ldots x_0$ จากการที่เลขฐานสองของ x และ 0 แตกต่างกัน ทำให้สามารถสร้างเซต $H_x = \{i: 0 \leq i \leq k\text{-}1$ และ$x_i = 1\}$ ซึ่งเป็นเซตของตำแหน่งที่ x_i เท่ากับ 1 ได้ และจะสังเกตได้ว่าระยะทางแฮมมิ่ง (Hamming distance) ระหว่าง x และ 0 คือ $|\ H_x\ |$ นั่นเอง ยกตัวอย่างเช่น กำหนดให้โหนด x = 101(5) ซึ่งเป็นสมาชิกของ Q_3 จะเห็นได้ว่าเลขฐานสองของโหนด 101(5) และโหนด 000(0) แตกต่างกันตรงตำแหน่งที่ 0 และ 2 ดังนั้นจะได้ $H_5 = \{0, 2\}$ และระยะทางแฮมมิ่งระหว่างโหนด 5 และ 0 $= |\ H_5\ | = 2$

จัตุรัสลาตินของระยะทางแฮมมิ่ง (Hamming Distance Latin Square: HDLS) [13] คือ เมตริกซ์ที่สมาชิกในแถวแรกทั้งหมดเป็นสมาชิกของ H_x สมมติให้โหนด $x(\neq 0)$ ใดๆ เป็นสมาชิกของ Q_k มีระยะทางแฮมมิ่งเมื่อเทียบกับโหนด 0 เท่ากับ t จะทำให้ได้ $H_x = \{i_0, i_1, \ldots, i_{t-2}, i_{t-1}\}$ โดยที่ $i_0 < i_1 < \ldots < i_{t-2} < i_{t-1}$ และจะสามารถสร้างจัตุรัสลาตินของระยะทางแฮมมิ่งได้ 2 แบบคือ จัตุรัสลาตินของระยะทางแฮมมิ่งที่มีการหมุนรอบแบบเพิ่ม (Increasingly rotational HDLS) และจัตุรัสลาตินของระยะทางแฮมมิ่งที่มีการหมุนรอบแบบลด (Decreasingly rotational HDLS) ซึ่งจัตุรัสลาตินของระยะทางแฮมมิ่งทั้ง 2 แบบ แสดงเป็นเมตริกซ์ดังต่อไปนี้

$$I_k(x)=\begin{bmatrix} i_0 & i_1 & \cdots & i_{t-2} & i_{t-1} \\ i_1 & i_2 & \cdots & i_{t-1} & i_0 \\ \vdots & & \cdots & & \\ i_{t-2} & i_{t-1} & \cdots & i_{t-4} & i_{t-3} \\ i_{t-1} & i_0 & \cdots & i_{t-3} & i_{t-2} \end{bmatrix} \text{ และ } D_k(x)=\begin{bmatrix} i_{t-1} & i_{t-2} & \cdots & i_1 & i_0 \\ i_{t-2} & i_{t-3} & \cdots & i_0 & i_{t-1} \\ \vdots & & \cdots & & \\ i_1 & i_0 & \cdots & i_3 & i_2 \\ i_0 & i_{t-1} & \cdots & i_2 & i_1 \end{bmatrix}$$

เมตริกซ์ $I_k(x)$ คือ จัตุรัสลาตินของระยะทางแฮมมิ่งที่มีการหมุนรอบแบบเพิ่ม ส่วนเมตริกซ์ $D_k(x)$ คือ จัตุรัสลาตินของระยะทางแฮมมิ่งที่มีการหมุนรอบแบบลด โดยที่สมาชิกตัวที่ j+1 หรือ i_{j+1} ใน $I_k(x)$ จะเรียกว่า ลูก (Successor) ของ i_j (หากพิจารณา $D_k(x)$ สมาชิกตัวที่ j-1 หรือ i_{j-1} จะเป็นลูกของ i_j) แสดงโดยใช้สัญลักษณ์ $succ(i_j)$ และสมาชิกตัวที่ j-1 หรือ i_{j-1} ใน $I_k(x)$ จะเรียกว่า พ่อแม่ (Predecessor) ของ i_j (สำหรับ $D_k(x)$ สมาชิกตัวที่ j+1 หรือ i_{j+1} จะเป็นพ่อแม่ของ i_j) แสดงโดยใช้สัญลักษณ์ $pred(i_j)$ โดยเมตริกซ์ $I_4(13)$ และ $D_4(13)$ แสดงดังตัวอย่างที่ 2.1

ตัวอย่างที่ 2.1 พิจารณาที่โหนด x = 1101(13) ใน Q_4 จะสามารถสร้าง H_{13} = {0, 2, 3} และจะทำให้สร้างเมตริกซ์จัตุรัสลาตินของระยะทางแฮมมิ่งทั้ง 2 แบบได้ดังต่อไปนี้

$$I_4(13)=\begin{bmatrix} 0 & 2 & 3 \\ 2 & 3 & 0 \\ 3 & 0 & 2 \end{bmatrix} \text{ และ } D_4(13)=\begin{bmatrix} 3 & 2 & 0 \\ 2 & 0 & 3 \\ 0 & 3 & 2 \end{bmatrix}$$

หากพิจารณา i_j = 2 จะพบว่าลูกของ 2 หรือ succ(2) = 3 และพ่อแม่ของ 2 หรือ pred(2) = 0

และเมื่อพิจารณาโหนด x = 1111(15) ใน Q_4 จะสามารถสร้าง H_{15} = {0, 1, 2, 3} และจะทำให้สร้าง $I_4(15)$ และ $D_4(15)$ ได้ดังเมตริกซ์ต่อไปนี้

$$I_4(15)=\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix} \text{ และ } D_4(15)=\begin{bmatrix} 3 & 2 & 1 & 0 \\ 2 & 1 & 0 & 3 \\ 1 & 0 & 3 & 2 \\ 0 & 3 & 2 & 1 \end{bmatrix}$$

หากพิจารณา i_j = 2 จะพบว่าลูกของ 2 หรือ succ(2) = 3 และพ่อแม่ของ 2 หรือ pred(2) = 1

ซึ่งหลักการของจัตุรัสลาตินของระยะทางแฮมมิ่งนี้สามารถนำมาประยุกต์ใช้สำหรับหาเส้นทางที่ไม่ซ้ำกันในแต่ละ T_i ($0 \le i \le k-1$) ของ Q_k ด้วยวิธีการแบบขนานได้

2.5.2 ขั้นตอนวิธีสำหรับสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนด 0 บนเครือข่ายแบบไฮเปอร์คิว

เนื้อหาในส่วนนี้จะกล่าวถึงวิธีการสร้างชุดของต้นไม้ทอดข้ามที่แตกต่างบน Q_k หรือ $\tau = \{T_i: 0 \leq i \leq k-1\}$ ที่มีโหนดรากเป็นโหนด 0 โดยการให้ทุกๆ โหนด (หน่วยประมวลผล) $x(\neq 0)$ ในระบบเครือข่ายคำนวณหาโหนดพ่อแม่ในแต่ละ T_i ตั้งแต่ T_0 ถึง T_{k-1} ซึ่งสัญลักษณ์ที่ใช้แทนเส้นทางของ Q_k คือ $x \xrightarrow{+2^i} y$ ในกรณีที่ $x + 2^i = y$ หรือ $x \xrightarrow{-2^i} y$ ในกรณีที่ $x - 2^i = y$ ซึ่งมีความหมายว่า x และ y เป็นโหนดที่อยู่ติดกันในเส้นทางนั้นๆ

```
Algorithm GEN-PARENTS
begin
for all x(≠0)∈Qk with binary string x=xk-1xk-2 ...x0 pardo
   for i = 0 to k-1 do
      if (xi=0) parent(Ti,x) = x+2^i; // adjust bit weight at xi
      if (xi=1) parent(Ti,x) = x-2^succ(i); // apply HDLS to adjust bit weight at xsucc(i)
   end for i
end for all
end GEN-PARENTS
```

ขั้นตอนวิธีดังกล่าวสามารถสร้างต้นไม้ทอดข้ามที่แตกต่างจำนวน k ต้น บน Q_k ซึ่งมีโหนดรากเป็นโหนด 0 ด้วยความซับซ้อนด้านเวลาเท่ากับ O(k) โดยจะแสดงขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง 4 ต้น ดังรูปที่ 2.7 ได้แก่ T_0, T_1, T_2 และ T_3 บน Q_4 ดังตัวอย่างที่ 2.2 และข้อมูลทั้งหมดสำหรับการสร้าง T_0, T_1, T_2 และ T_3 บน Q_4 แสดงดังตารางที่ 2.1, 2.2, 2.3 และ 2.4 ตามลำดับ

<u>ตัวอย่างที่ 2.2</u> ขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง 4 ต้น ได้แก่ T_0, T_1, T_2 และ T_3 บน Q_4 กำหนดให้โหนด $x(\neq 0)$ มีหมายเลขประจำโหนดในรูปแบบของเลขฐานสองเป็น $x = x_3x_2x_1x_0$

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 0)$ ใน T_0 บน Q_4

<u>กรณีที่ 1</u> ถ้า $x_0 = 0$ จะได้ $x \in \{2, 4, 6, 8, 10, 12, 14\}$

ดังนั้น $parent(T_0, x) = x + 2^0 = x + 1$ จะได้ $parent(T_0, x) = \{3, 5, 7, 9, 11, 13, 15\}$

<u>กรณีที่ 2</u> ถ้า $x_0 = 1$ จะแบ่งได้หลายกรณีดังนี้

$x \in \{1\}$ ดังนั้น $succ(0) = 0$ จะได้ $parent(T_0, x) = x - 2^0 = 1 - 1 = 0$

$x \in \{3, 7, 11, 15\}$ ดังนั้น $succ(0) = 1$ จะได้ $parent(T_0, x) = x - 2^1 = x - 2$

ดังนั้น parent(T_0, x) = {1, 5, 9, 13}

x ∈ {5, 13} ดังนั้น succ(0) = 2 จะได้ parent(T_0, x) = x - 2^2 = x – 4

ดังนั้น parent(T_0, x) = {1, 9}

x ∈ {9} ดังนั้น succ(0) = 3 จะได้ parent(T_0, x) = x - 2^3 = 9 - 8 = 1

ตารางที่ 2.1 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_4 (i = 0)

x	$x_3x_2x_1x_0$	succ(0)	$2^{succ(0)}$	parent(T_0, x)
1	000***1***	0	1	000***0*** (0)
2	001***0***	-	-	001***1*** (3)
3	00***1***1	1	2	00***0***1 (1)
4	010***0***	-	-	010***1*** (5)
5	0***1***01	2	4	0***0***01 (1)
6	011***0***	-	-	011***1*** (7)
7	01***1***1	1	2	01***0***1 (5)
8	100***0***	-	-	100***1*** (9)
9	***1***001	3	8	***0***001 (1)
10	101***0***	-	-	101***1*** (11)
11	10***1***1	1	2	10***0***1 (9)
12	110***0***	-	-	110***1*** (13)
13	1***1***01	2	4	1***0***01 (9)
14	111***0***	-	-	111***1*** (15)
15	11***1***1	1	2	11***0***1 (13)

$I_4(1) = [\,(0)\,]$

$I_4(3) = \begin{bmatrix} 0 & (1) \\ 1 & 0 \end{bmatrix}$

$I_4(5) = \begin{bmatrix} 0 & (2) \\ 2 & 0 \end{bmatrix}$

$I_4(9) = \begin{bmatrix} 0 & (3) \\ 3 & 0 \end{bmatrix}$

$I_4(7) = \begin{bmatrix} 0 & (1) & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

$I_4(11) = \begin{bmatrix} 0 & (1) & 3 \\ 1 & 3 & 0 \\ 3 & 0 & 1 \end{bmatrix}$

$I_4(13) = \begin{bmatrix} (0) & 2 & 3 \\ 2 & 3 & 0 \\ 3 & 0 & 2 \end{bmatrix}$

$I_4(15) = \begin{bmatrix} 0 & (1) & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด x(≠0) ใน T_1 บน Q_4

<u>กรณีที่ 1</u> ถ้า $x_1 = 0$ จะได้ x ∈ {1, 4, 5, 8, 9, 12, 13}

ดังนั้น parent(T_1, x) = x + 2^1 = x + 2 จะได้ parent(T_1, x) = {3, 6, 7, 10, 11, 14, 15}

<u>กรณีที่ 2</u> ถ้า $x_1 = 1$ จะแบ่งได้หลายกรณีดังนี้

x ∈ {3} ดังนั้น succ(1) = 0 จะได้ parent(T_1, x) = x - 2^0 = 3 - 1 = 2

x ∈ {2} ดังนั้น succ(1) = 1 จะได้ parent(T_1, x) = x - 2^1 = 2 – 2 = 0

$x \in \{6, 7, 14, 15\}$ ดังนั้น succ(1) = 2 จะได้ $parent(T_1, x) = x - 2^2 = x - 4$

ดังนั้น $parent(T_1, x) = \{2, 3, 10, 11\}$

$x \in \{10, 11\}$ ดังนั้น succ(1) = 3 จะได้ $parent(T_1, x) = x - 2^3 = x - 8$

ดังนั้น $parent(T_1, x) = \{2, 3\}$

ตารางที่ 2.2 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_4 (i = 1)

x	$x_3x_2x_1x_0$	succ(1)	$2^{succ(1)}$	$parent(T_1, x)$
1	000***1***	-	-	00***1***1 (3)
2	00***1***0	1	2	000***0*** (0)
3	00***1***1	0	1	00***1***0 (2)
4	0***1***00	-	-	0***1***10 (6)
5	0***1***01	-	-	0***1***11 (7)
6	0***1***10	2	4	0***0***10 (2)
7	0***1***11	2	4	0***0***11 (3)
8	1***0***00	-	-	1***0***10 (10)
9	1***0***01	-	-	1***0***11 (11)
10	***1***010	3	8	***0***010 (2)
11	***1***011	3	8	***0***011 (3)
12	11***0***0	-	-	11***1***0 (14)
13	11***0***1	-	-	11***1***1 (15)
14	1***1***10	2	4	1***0***10 (10)
15	1***1***11	2	4	1***0***11 (11)

$I_4(2) = [\,(1)\,]$

$I_4(3) = \begin{bmatrix} 0 & 1 \\ 1 & (0) \end{bmatrix}$

$I_4(6) = \begin{bmatrix} 1 & (2) \\ 2 & 1 \end{bmatrix}$

$I_4(10) = \begin{bmatrix} 1 & (3) \\ 3 & 1 \end{bmatrix}$

$I_4(7) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & (2) & 0 \\ 2 & 0 & 1 \end{bmatrix}$

$I_4(11) = \begin{bmatrix} 0 & 1 & 3 \\ 1 & (3) & 0 \\ 3 & 0 & 1 \end{bmatrix}$

$I_4(14) = \begin{bmatrix} (1) & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$

$I_4(15) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & (2) & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 0)$ ใน T_2 บน Q_4

<u>กรณีที่ 1</u> ถ้า $x_2 = 0$ จะได้ $x \in \{1, 2, 3, 8, 9, 10, 11\}$

ดังนั้น $parent(T_2, x) = x + 2^2 = x + 4$ จะได้ $parent(T_2, x) = \{5, 6, 7, 12, 13, 14, 15\}$

<u>กรณีที่ 2</u> ถ้า $x_2 = 1$ จะแบ่งได้หลายกรณีดังนี้

$x \in \{5, 7\}$ ดังนั้น succ(2) = 0 จะได้ $parent(T_2, x) = x - 2^0$

ดังนั้น $parent(T_2, x) = \{2, 3\}$

$x \in \{6\}$ ดังนั้น succ(2) = 1 จะได้ $parent(T_2, x) = x - 2^1 = 6 - 2 = 4$

$x \in \{4\}$ ดังนั้น succ(2) = 2 จะได้ $parent(T_2, x) = x - 2^2 = 4 - 4 = 0$

$x \in \{12, 13, 14, 15\}$ ดังนั้น succ(2) = 3 จะได้ $parent(T_2, x) = x - 2^3 = x - 8$

ดังนั้น $parent(T_2, x) = \{4, 5, 6, 7\}$

ตารางที่ 2.3 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_4 (i = 2)

x	$x_3x_2x_1x_0$	succ(2)	$2^{succ(2)}$	parent(T_2, x)
1	0***0***01	-	-	0***1***01 (5)
2	0***0***10	-	-	0***1***10 (6)
3	0***0***11	-	-	0***1***11 (7)
4	0***1***00	2	4	0***0***00 (0)
5	010***1***	0	1	010***0*** (4)
6	01***1***0	1	2	01***0***0 (4)
7	011***1***	0	1	011***0*** (6)
8	1***0***00	-	-	1***1***00 (12)
9	1***0***01	-	-	1***1***01 (13)
10	1***0***10	-	-	1***1***10 (14)
11	1***0***11	-	-	1***1***11 (15)
12	***1***100	3	8	***0***100 (4)
13	***1***101	3	8	***0***101 (5)
14	***1***110	3	8	***0***110 (6)
15	***1***111	3	8	***0***111 (7)

$I_4(4) = [\,(2)\,]$

$I_4(5) = \begin{bmatrix} 0 & 2 \\ 2 & (0) \end{bmatrix}$

$I_4(6) = \begin{bmatrix} 1 & 2 \\ 2 & (1) \end{bmatrix}$

$I_4(12) = \begin{bmatrix} 2 & (3) \\ 3 & 2 \end{bmatrix}$

$I_4(7) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & (0) & 1 \end{bmatrix}$

$I_4(13) = \begin{bmatrix} 0 & 2 & 3 \\ 2 & (3) & 0 \\ 3 & 0 & 2 \end{bmatrix}$

$I_4(14) = \begin{bmatrix} 1 & 2 & 3 \\ 2 & (3) & 1 \\ 3 & 1 & 2 \end{bmatrix}$

$I_4(15) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & (3) & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 0)$ ใน T_3 บน Q_4

<u>กรณีที่ 1</u> ถ้า $x_3 = 0$ จะได้ $x \in \{1, 2, 3, 4, 5, 6, 7\}$

ดังนั้น $parent(T_3, x) = x + 2^3 = x + 8$ จะได้ $parent(T_3, x) = \{9, 10, 11, 12, 13, 14, 15\}$

<u>กรณีที่ 2</u> ถ้า $x_3 = 1$ จะแบ่งได้หลายกรณีดังนี้

$x \in \{9, 11, 13, 15\}$ ดังนั้น succ(3) = 0 จะได้ $parent(T_3, x) = x - 2^0$

ดังนั้น $parent(T_3, x) = \{8, 10, 12, 14\}$

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

$x \in \{10, 14\}$ ดังนั้น succ(3) = 1 จะได้ $parent(T_3, x) = x - 2^1$

ดังนั้น $parent(T_3, x) = \{8, 12\}$

$x \in \{12\}$ ดังนั้น succ(3) = 2 จะได้ $parent(T_3, x) = x - 2^2 = 12 - 4 = 8$

$x \in \{8\}$ ดังนั้น succ(3) = 3 จะได้ $parent(T_3, x) = x - 2^3 = 8 - 8 = 0$

ตารางที่ 2.4 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_3 บน Q_4 (i = 3)

x	$x_3x_2x_1x_0$	succ(3)	$2^{succ(3)}$	parent(T_3, x)
1	***0***001	-	-	***1***001 (9)
2	***0***010	-	-	***1***010 (10)
3	***0***011	-	-	***1***011 (11)
4	***0***100	-	-	***1***100 (12)
5	***0***101	-	-	***1***101 (13)
6	***0***110	-	-	***1***110 (14)
7	***0***111	-	-	***1***111 (15)
8	***1***000	3	8	***0***000 (0)
9	100***1***	0	1	100***0*** (8)
10	10***1***0	1	2	10***0***0 (8)
11	101***1***	0	1	101***0*** (10)
12	1***1***00	2	4	1***0***00 (8)
13	110***1***	0	1	110***0*** (12)
14	11***1***0	1	2	11***0***0 (12)
15	1111	0	1	111***0*** (14)

$I_4(8) = [\,(3)\,]$

$I_4(9) = \begin{bmatrix} 0 & 3 \\ 3 & (0) \end{bmatrix}$

$I_4(10) = \begin{bmatrix} 1 & 3 \\ 3 & (1) \end{bmatrix}$

$I_4(12) = \begin{bmatrix} 2 & 3 \\ 3 & (2) \end{bmatrix}$

$I_4(11) = \begin{bmatrix} 0 & 1 & 3 \\ 1 & 3 & 0 \\ 3 & (0) & 1 \end{bmatrix}$

$I_4(13) = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 3 & 0 \\ 3 & (0) & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(3)

$I_4(14) = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & (1) & 2 \end{bmatrix}$

$I_4(15) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & (0) & 1 & 2 \end{bmatrix}$

2.6 การพิสูจน์ความถูกต้องของขั้นตอนวิธี

ในส่วนนี้จะเป็นการพิสูจน์ความถูกต้องของขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานบนเครือข่ายแบบไฮเปอร์คิวโดยจะทำการพิสูจน์ใน 3 ทฤษฎีบทด้วยกันได้แก่ ทฤษฎีบทที่ 1 จะเป็นการพิสูจน์ว่ามีต้นไม้ทอดข้ามที่แตกต่าง T_i ($0 \leq i \leq k-1$) ซึ่งมีโหนดรากเป็นโหนด 0 จำนวน k ต้นสำหรับเครือข่ายแบบไฮเปอร์คิว k มิติ และหากนับจำนวนเส้น

105312

เชื่อมบนเส้นทางเป็นระยะห่างระหว่างโหนด ทฤษฎีบทที่ 2 เป็นการพิสูจน์ว่าความสูงของต้นไม้ทอดข้ามที่แตกต่าง k ต้นที่มีโหนดรากเป็นโหนด 0 มีความสูงไม่เกิน k+1 เส้นเชื่อม และในทฤษฎีบทที่ 3 จะเป็นการพิสูจน์ว่าเส้นทางจากทุกๆ โหนด $x(\neq 0)$ ซึ่งเป็นสมาชิกของเครือข่ายแบบไฮเปอร์คิว k มิติไปยังโหนด 0 ในแต่ละ T_i $(0 \leq i \leq k\text{-}1)$ เป็นเส้นทางที่แตกต่างกัน สำหรับการพิจารณาในแต่ละทฤษฎีบทกำหนดให้หมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ $x = x_{k-1}x_{k-2}\cdots x_0$

ทฤษฎีบทที่ 1 สำหรับเครือข่ายแบบไฮเปอร์คิว k มิติใดๆ จะมีต้นไม้ทอดข้ามที่แตกต่าง T_i $(0 \leq i \leq k\text{-}1)$ ซึ่งมีโหนดรากเป็นโหนด 0 จำนวน k ต้น

พิสูจน์ จากคุณสมบัติการเชื่อมต่อระหว่างโหนดของ Q_k ที่ว่า โหนดแต่ละโหนดบนไฮเปอร์คิวจะมีเส้นเชื่อมต่อกับโหนดประชิดในแต่ละมิติทั้งหมด k เส้น ดังนั้นโหนด 0 จึงมีโหนดประชิดทั้งหมด k โหนด ได้แก่ โหนด $2^0, 2^1, \ldots, 2^{k-1}$ โดยจะคำนวณหาโหนดลูกของโหนด 0 ได้จากการปรับบิตของโหนด 0 ทั้ง k บิตตั้งแต่บิตที่ 0 จนถึงบิต k-1 จาก 0 เป็น 1 ซึ่งก็คือโหนด 2^i $(0 \leq i \leq k\text{-}1)$ ในแต่ละ T_i ตั้งแต่ T_0, T_1 จนถึง T_{k-1} ตามลำดับ ซึ่งแสดงว่าในแต่ละ T_i โหนดลูกของโหนด 0 จะต้องแตกต่างกันและมีเพียงโหนดเดียวเท่านั้น จากเหตุผลดังกล่าวข้างต้นจึงพิสูจน์ได้ว่าบนเครือข่ายแบบไฮเปอร์คิว k มิติใดๆ จะมีต้นไม้ทอดข้ามที่แตกต่าง T_i $(0 \leq i \leq k\text{-}1)$ ซึ่งมีโหนดรากเป็นโหนด 0 จำนวน k ต้น □

ตัวอย่างของต้นไม้ทอดข้ามที่แตกต่าง 4 ต้นที่มีโหนดรากเป็นโหนด 0 ได้แก่ T_0, T_1, T_2 และ T_3 ของ Q_4 แสดงดังรูปที่ 2.7 และตัวอย่างการคำนวณหาโหนดลูกของโหนด 0 ใน T_0 ถึง T_3 ของ Q_4 แสดงดังตารางที่ 2.5

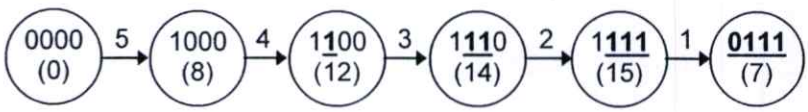
ตารางที่ 2.5 แสดงตัวอย่างการคำนวณหาโหนดลูกของโหนด 0 ตั้งแต่ T_0 ถึง T_3 ของ Q_4

T_i	หมายเลขประจำโหนดของโหนด 0 (รูปแบบเลขฐานสอง)	โหนดลูกของโหนด 0 (รูปแบบเลขฐานสอง)	โหนดลูกของโหนด 0 (การปรับบิตที่ i ของ 0)
T_0	000***0***	000***1***	1 $(=2^0)$
T_1	00***0***0	00***1***0	2 $(=2^1)$
T_2	0***0***00	0***1***00	4 $(=2^2)$
T_3	***0***000	***1***000	8 $(=2^3)$

ทฤษฎีบทที่ 2 สำหรับเครือข่ายไฮเปอร์คิว k มิติใดๆ ความสูงของต้นไม้ทอดข้ามที่แตกต่าง T_i ($0 \leq i \leq k-1$) ซึ่งมีโหนดรากเป็นโหนด 0 เป็นความสูงที่เหมาะสมคือมีความสูงไม่เกิน k+1 เส้นเชื่อม

พิสูจน์ เนื่องจากความยาวของเส้นทางที่แตกต่างจากโหนด $x \notin \{0, 2^i\}$ ไปยังโหนด 2^i เท่ากับระยะทางแฮมมิ่งระหว่าง x กับ 2^i ดังนั้นเส้นทางที่ยาวที่สุดจากโหนด $x = 11..1x_i1..1 \notin \{0, 2^i\}$ โดยที่ $x_i = 0$ ไปยังโหนด 2^i จึงเท่ากับ k เส้นเชื่อม และเนื่องจากระยะทางแฮมมิ่งระหว่างโหนด 0 และโหนด 2^i เท่ากับ 1 เส้นเชื่อม ดังนั้นเส้นทางจากโหนด 0 จนถึงโหนด $x = 11..1x_i1..1\ 1$ ซึ่งเป็นโหนดที่อยู่ไกลจากโหนด 0 ที่สุดในเส้นทางจึงเท่ากับ k+1 เส้นเชื่อม จากเหตุผลข้างต้นจึงพิสูจน์ได้ว่า ความสูงของต้นไม้ทอดข้ามที่แตกต่าง T_i ($0 \leq i \leq k-1$) ที่สร้างจากขั้นตอนวิธีเป็นความสูงที่เหมาะสมคือมีความสูงไม่เกิน k+1 เส้นเชื่อม □

ตัวอย่างที่ 2.3 พิจารณาเส้นทางที่ยาวที่สุดใน T_3 (ดังรูปที่ 2.8) ที่มีโหนดรากเป็นโหนด 0 บน Q_4 (k = 4, i = 3, r = 0)



รูปที่ 2.8 แสดงเส้นทางที่ยาวที่สุดใน T_3 ที่มีโหนดรากเป็นโหนด 0 บน Q_4

เมื่อพิจารณาบน T_3 จะพบว่าโหนดลูกของโหนด 0 ใน T_3 คือโหนด 8(1000) และโหนดที่ไกลที่สุดในเส้นทางคือโหนด $7 \notin \{0, 8\}$ เมื่อพิจารณาหมายเลขประจำโหนดของโหนด 7 ในรูปแบบของเลขฐานสองจะได้ว่า $x = x_3x_2x_1x_0 = 0111$ โดยที่ $x_3 = 0$. จากเลขฐานสองซึ่งเป็นหมายเลขประจำโหนดของทั้งโหนด 7(0111) และโหนด 8(1000) จะพบว่าระยะทางแฮมมิ่งระหว่างโหนดทั้ง 2 นี้เท่ากับ 4 (= k) หรืออีกนัยหนึ่งก็คือเส้นทางระหว่าง 2 โหนดนี้คือ 4 เส้นเชื่อม (= k) และเนื่องจากระยะทางระหว่างโหนด 0 และโหนด 8 เท่ากับ 1 เส้นเชื่อม ทำให้ระยะทางจากโหนด 7 ไปยังโหนด 0 เท่ากับ 5 (= k+1) เส้นเชื่อม

ทฤษฎีบทที่ 3 ในต้นไม้ทอดข้ามที่แตกต่าง T_i ($0 \leq i \leq k-1$) ที่มีโหนดรากเป็นโหนด 0 เส้นทางจากโหนด $x(\neq 0)$ ซึ่งเป็นสมาชิกของเครือข่ายแบบไฮเปอร์คิว k มิติไปยังโหนด 0 เป็นเส้นทางที่แตกต่างกัน

พิสูจน์ จากขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างบนเครือข่ายแบบไฮเปอร์คิว k มิติ ซึ่งประยุกต์ใช้จัตุรัสลาตินของระยะทางแฮมมิ่งเพื่อช่วยหาเส้นทางที่ไม่ซ้ำกันจากโหนด $x(\neq 0) \in Q_k$ ไปยังโหนดราก ($r = 0$) โดยพิจารณาเลขประจำโหนดในรูปแบบของเลขฐานสองของ $x = x_{k-1}x_{k-2}\ldots x_0$ โดยที่ $H_x = \{i_0, i_1, \ldots, i_{t-2}, i_{t-1}\}$ และ $i_0 < i_1 < \ldots < i_{t-2} < i_{t-1}$ โดยสัญลักษณ์ที่ใช้แทนเส้นทางของ Q_k คือ $x \xrightarrow{+2^i} y$ ในกรณีที่ $x + 2^i = y$ หรือ $x \xrightarrow{-2^i} y$ ในกรณีที่ $x - 2^i = y$ ซึ่งมีความหมายว่า x และ y เป็นโหนดที่อยู่ติดกันในเส้นทางนั้นๆ ในที่นี้สามารถพิจารณาเส้นทางที่แตกต่างใน T_i ได้ 2 กรณีดังนี้

กรณีที่ 1 : $i \in H_x$ (หรือกรณีที่ $x_i = 1$) ในที่นี้สมมติให้ $i = i_p$ สำหรับ p ใดๆ โดยที่ $p = 0, 1, \ldots, t-1$ สืบเนื่องจากกฎที่ว่า x เป็นโหนดประชิดของ $x - 2^{i_p+1}$ ใน T_i จึงกำหนดให้ $y = x - 2^{i_p+1}$ และเนื่องจาก $H_y = \{i_0, i_1, \ldots, i_p, i_{p+2}, \ldots, i_{t-1}\}$ ดังนั้น y จึงเป็นโหนดประชิดของ $y - 2^{i_p+2}$ ใน T_i และกำหนดให้ $z = y - 2^{i_p+2}$ โดยที่ $H_z = \{i_0, i_1, \ldots, i_p, i_{p+3}, \ldots, i_{t-1}\}$ ดังนั้น z จึงเป็นโหนดประชิดของ $z - 2^{i_p+3}$ ใน T_i ซึ่งทำให้สามารถหาเส้นทางที่แตกต่างที่เชื่อมระหว่าง x และ 0 ใน T_i ได้ดังนี้

$$x \xrightarrow{-2^{i_p+1}} (x-2^{i_p+1}) \xrightarrow{-2^{i_p+2}} (x-2^{i_p+1}-2^{i_p+2}) \xrightarrow{-2^{i_p+3}} \ldots \xrightarrow{-2^{i_p-1}} (2^{i_p}) \xrightarrow{-2^{i_p}} 0$$

กรณีที่ 2 : $i \notin H_x$ (หรือกรณีที่ $x_i = 0$) จากกฎที่ว่า x เป็นโหนดประชิดของ $x + 2^i$ ใน T_i กำหนดให้ $y = x + 2^i$ ซึ่งทำให้ $i \in H_x$ จากกรณีที่ 1 ที่แสดงว่ามีเส้นทางที่แตกต่าง P ที่เชื่อมระหว่าง y และ 0 ใน T_i ดังนั้น $x \xrightarrow{+2^i} y$ จึงเป็นส่วนหนึ่งของเส้นทางที่แตกต่าง P ซึ่งทำให้ x เชื่อมกับ 0 ใน T_i

จากการพิจารณาเส้นทางที่แตกต่างใน T_i ทั้ง 2 กรณีข้างต้น จะเห็นได้ว่าทุกๆ โหนดมีค่า x หรือมีหมายเลขประจำโหนดที่แตกต่างกัน และเมื่อพิจารณา ณ ค่า i ใดๆ เส้นทางจากทุกๆ โหนด $x(\neq 0)$ ไปยังโหนด 0 ใน T_i ย่อมแตกต่างกัน และจากค่า i ที่พิจารณาในแต่ละ T_i ไม่เหมือนกัน ($i \in \{0, 1, 2, \ldots, k-1\}$) เส้นทางจากโหนด $x(\neq 0)$ ใดๆ ไปยังโหนด 0 จึงเป็นเส้นทางที่ไม่ซ้ำกัน และสรุปได้ว่าเส้นทางจากทุกๆ โหนด $x(\neq 0)$ ไปยังโหนด 0 ในทุกๆ T_i เป็นเส้นทางที่แตกต่างกัน □

ตัวอย่างที่ 2.4 พิจารณาเส้นทางที่แตกต่างจากทุกๆ โหนดตั้งแต่โหนด 0001(1) ถึงโหนด 1111(15) บน Q_4 ไปยังโหนด 0 ในต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ได้ดังนี้

เส้นทางจาก x = 0001(1) ไปยัง r = 0000(0)

$T_0[1, 0]$: $1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[1, 0]$: $1{:}0001 \xrightarrow{+2^1} 3{:}0011 \xrightarrow{-2^0} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[1, 0]$: $1{:}0001 \xrightarrow{+2^2} 5{:}0101 \xrightarrow{-2^0} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[1, 0]$: $1{:}0001 \xrightarrow{+2^3} 9{:}1001 \xrightarrow{-2^0} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 0010(2) ไปยัง r = 0000(0)

$T_0[2, 0]$: $2{:}0010 \xrightarrow{+2^0} 3{:}0011 \xrightarrow{-2^1} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[2, 0]$: $2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[2, 0]$: $2{:}0010 \xrightarrow{+2^2} 6{:}0110 \xrightarrow{-2^1} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[2, 0]$: $2{:}0010 \xrightarrow{+2^3} 10{:}1010 \xrightarrow{-2^1} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 0011(3) ไปยัง r = 0000(0)

$T_0[3, 0]$: $3{:}0011 \xrightarrow{-2^1} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[3, 0]$: $3{:}0011 \xrightarrow{-2^0} 2{:}0011 \xrightarrow{-2^1} 0{:}0000$

$T_2[3, 0]$: $3{:}0011 \xrightarrow{+2^2} 7{:}0111 \xrightarrow{-2^0} 6{:}0110 \xrightarrow{-2^1} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[3, 0]$: $3{:}0011 \xrightarrow{+2^3} 11{:}1011 \xrightarrow{-2^0} 10{:}1010 \xrightarrow{-2^1} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 0100(4) ไปยัง r = 0000(0)

$T_0[4, 0]$: $4{:}0100 \xrightarrow{+2^0} 5{:}0101 \xrightarrow{-2^2} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[4, 0]$: $4{:}0100 \xrightarrow{+2^1} 6{:}0110 \xrightarrow{-2^2} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[4, 0]$: $4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[4, 0]$: $4{:}0100 \xrightarrow{+2^3} 12{:}1100 \xrightarrow{-2^2} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 0101(5) ไปยัง r = 0000(0)

$T_0[5, 0]$: $5{:}0101 \xrightarrow{-2^2} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[5, 0]$: $5{:}0101 \xrightarrow{+2^1} 7{:}0111 \xrightarrow{-2^2} 3{:}0011 \xrightarrow{-2^0} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[5, 0]$: $5{:}0101 \xrightarrow{-2^0} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[5, 0]$: $5{:}0101 \xrightarrow{+2^3} 13{:}1101 \xrightarrow{-2^0} 12{:}1100 \xrightarrow{-2^2} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 0110(6) ไปยัง r = 0000(0)

$T_0[6, 0]$: $6{:}0110 \xrightarrow{+2^0} 7{:}0111 \xrightarrow{-2^1} 5{:}0101 \xrightarrow{-2^2} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[6, 0]$: $6{:}0110 \xrightarrow{-2^2} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[6, 0]$: $6{:}0110 \xrightarrow{-2^1} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[6, 0]$: $6{:}0110 \xrightarrow{+2^3} 14{:}1110 \xrightarrow{-2^1} 12{:}1100 \xrightarrow{-2^2} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 0111(7) ไปยัง r = 0000(0)

$T_0[7, 0]$: $7{:}0111 \xrightarrow{-2^1} 5{:}0101 \xrightarrow{-2^2} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[7, 0]$: $7{:}0111 \xrightarrow{-2^2} 3{:}0011 \xrightarrow{-2^0} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[7, 0]$: $7{:}0111 \xrightarrow{-2^0} 6{:}0110 \xrightarrow{-2^1} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[7, 0]$: $7{:}0111 \xrightarrow{+2^3} 15{:}1111 \xrightarrow{-2^0} 14{:}1110 \xrightarrow{-2^1} 12{:}1100 \xrightarrow{-2^2} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 1000(8) ไปยัง r = 0000(0)

$T_0[8, 0]$: $8{:}1000 \xrightarrow{+2^0} 9{:}1001 \xrightarrow{-2^3} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[8, 0]$: $8{:}1000 \xrightarrow{+2^1} 10{:}1010 \xrightarrow{-2^3} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[8, 0]$: $8{:}1000 \xrightarrow{+2^2} 12{:}1100 \xrightarrow{-2^3} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[8, 0]$: $8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 1001(9) ไปยัง r = 0000(0)

$T_0[9, 0]$: $9{:}1001 \xrightarrow{-2^3} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[9, 0]$: $9{:}1001 \xrightarrow{+2^1} 11{:}1011 \xrightarrow{-2^3} 3{:}0011 \xrightarrow{-2^0} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[9, 0]$: $9{:}1001 \xrightarrow{+2^2} 13{:}1101 \xrightarrow{-2^3} 5{:}0101 \xrightarrow{-2^0} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[9, 0]$: $9{:}1001 \xrightarrow{-2^0} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 1010(10) ไปยัง r = 0000(0)

$T_0[10, 0]$: $10{:}1010 \xrightarrow{+2^0} 11{:}1011 \xrightarrow{-2^1} 9{:}1001 \xrightarrow{-2^3} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[10, 0]$: $10{:}1010 \xrightarrow{-2^3} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[10, 0]$: $10{:}1010 \xrightarrow{+2^2} 14{:}1110 \xrightarrow{-2^3} 6{:}0110 \xrightarrow{-2^1} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[10, 0]$: $10{:}1010 \xrightarrow{-2^1} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 1011(11) ไปยัง r = 0000(0)

$T_0[11, 0]$: $11{:}1011 \xrightarrow{-2^1} 9{:}1001 \xrightarrow{-2^3} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[11, 0]$: $11{:}1011 \xrightarrow{-2^3} 3{:}0011 \xrightarrow{-2^0} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[11, 0]$: $11{:}1011 \xrightarrow{+2^2} 15{:}1111 \xrightarrow{-2^3} 7{:}0111 \xrightarrow{-2^0} 6{:}0110 \xrightarrow{-2^1} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[11, 0]$: $11{:}1011 \xrightarrow{-2^0} 10{:}1010 \xrightarrow{-2^1} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 1100(12) ไปยัง r = 0000(0)

$T_0[12, 0]$: $12{:}1100 \xrightarrow{+2^0} 13{:}1101 \xrightarrow{-2^2} 9{:}1001 \xrightarrow{-2^3} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[12, 0]$: $12{:}1100 \xrightarrow{+2^1} 14{:}1110 \xrightarrow{-2^2} 10{:}1010 \xrightarrow{-2^3} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[12, 0]$: $12{:}1100 \xrightarrow{-2^3} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[12, 0]$: $12{:}1100 \xrightarrow{-2^2} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 1101(13) ไปยัง r = 0000(0)

$T_0[13, 0]$: $13{:}1101 \xrightarrow{-2^2} 9{:}1001 \xrightarrow{-2^3} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[13, 0]$: $13{:}1101 \xrightarrow{+2^1} 15{:}1111 \xrightarrow{-2^2} 11{:}1011 \xrightarrow{-2^3} 3{:}0011 \xrightarrow{-2^0} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[13, 0]$: $13{:}1101 \xrightarrow{-2^3} 5{:}0101 \xrightarrow{-2^0} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[13, 0]$: $13{:}1101 \xrightarrow{-2^0} 12{:}1100 \xrightarrow{-2^2} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 1110(14) ไปยัง r = 0000(0)

$T_0[14, 0]$: $14{:}1110 \xrightarrow{+2^0} 15{:}1111 \xrightarrow{-2^1} 13{:}1101 \xrightarrow{-2^2} 9{:}1001 \xrightarrow{-2^3} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[14, 0]$: $14{:}1110 \xrightarrow{-2^2} 10{:}1010 \xrightarrow{-2^3} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[14, 0]$: $14{:}1110 \xrightarrow{-2^3} 6{:}0110 \xrightarrow{-2^1} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[14, 0]$: $14{:}1110 \xrightarrow{-2^1} 12{:}1100 \xrightarrow{-2^2} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

เส้นทางจาก x = 1111(15) ไปยัง r = 0000(0)

$T_0[15, 0]$: $15{:}1111 \xrightarrow{-2^1} 13{:}1101 \xrightarrow{-2^2} 9{:}1001 \xrightarrow{-2^3} 1{:}0001 \xrightarrow{-2^0} 0{:}0000$

$T_1[15, 0]$: $15{:}1111 \xrightarrow{-2^2} 11{:}1011 \xrightarrow{-2^3} 3{:}0011 \xrightarrow{-2^0} 2{:}0010 \xrightarrow{-2^1} 0{:}0000$

$T_2[15, 0]$: $15{:}1111 \xrightarrow{-2^3} 7{:}0111 \xrightarrow{-2^0} 6{:}0110 \xrightarrow{-2^1} 4{:}0100 \xrightarrow{-2^2} 0{:}0000$

$T_3[15, 0]$: $15{:}1111 \xrightarrow{-2^0} 14{:}1110 \xrightarrow{-2^1} 12{:}1100 \xrightarrow{-2^2} 8{:}1000 \xrightarrow{-2^3} 0{:}0000$

บทที่ 3
การสร้างต้นไม้ทอดข้ามที่แตกต่างแบบใหม่ บนระบบเครือข่ายแบบไฮเปอร์คิว

วิทยานิพนธ์นี้เป็นการวิจัยที่เสนอขั้นตอนวิธีการแบบทั่วไปที่ใช้สร้างต้นไม้ทอดข้ามที่แตกต่าง k ต้นด้วยวิธีการแบบขนาน (Generalized Parallel Construction of Independent Spanning Tree) บนเครือข่ายแบบไฮเปอร์คิว k มิติ (k-dimensional hypercube) หรือ Q_k ซึ่งต้นไม้ทอดข้ามที่แตกต่าง k ต้นดังกล่าว มีโหนดรากเป็นโหนดใดๆ (r = 0, 1, 2, ..., หรือ 2^k-1) บนระบบเครือข่าย ซึ่งเป็นขั้นตอนวิธีที่ขยายแนวคิดจากขั้นตอนวิธีแบบขนานสำหรับสร้างต้นไม้ทอดข้ามที่แตกต่างซึ่งเฉพาะเจาะจงที่โหนดรากเป็นโหนด 0 [13] เพื่อให้ครอบคลุมการมีโหนดรากเป็นโหนดใดก็ได้ โดยวิธีใหม่นี้สามารถหาเส้นทางที่แตกต่างได้โดยใช้หลักการของจัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไป (Generalized Hamming Distance Latin Square : GHDLS) จากหลักการนี้ทำให้สร้างต้นไม้ทอดข้ามที่แตกต่างด้วยวิธีการแบบขนานได้ด้วยความซับซ้อนด้านเวลาเท่ากับ O(k)

เนื้อหาในบทที่ 3 นี้แบ่งออกเป็น 3 ส่วนดังต่อไปนี้ ส่วนที่ 1 การอธิบายความหมายของสัญลักษณ์ต่างๆ ที่ใช้ในขั้นตอนวิธี ส่วนที่ 2 ความรู้พื้นฐานที่จำเป็นสำหรับสร้างต้นไม้ทอดข้ามที่แตกต่างบน Q_k ส่วนที่ 3 ขั้นตอนวิธีการแบบขนานสำหรับการสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนดใดๆ (r = 0, 1, 2, ..., หรือ 2^k-1) บน Q_k

3.1 ความหมายของสัญลักษณ์ต่างๆ ที่ใช้ในขั้นตอนวิธี

ในการสร้างต้นไม้ทอดข้ามที่แตกต่าง (Independent Spanning Trees: ISTs) ที่มีโหนดรากเป็นโหนดใดๆ (r = 0, 1, 2, ..., หรือ 2^k-1) บน Q_k มีสัญลักษณ์ที่ใช้ในขั้นตอนวิธีดังต่อไปนี้

กำหนดให้ T_i แทนต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนด r ใดๆ เมื่อ $0 \leq i \leq k-1$

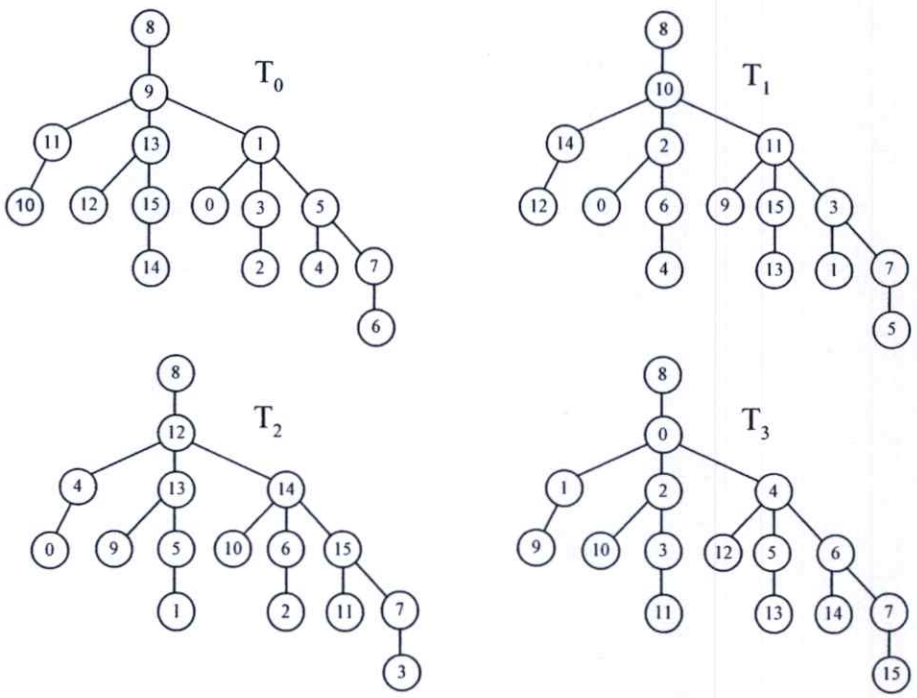
T[x, r] แทนเส้นทางจากโหนด x ถึงโหนดราก r บนต้นไม้ T

parent(T, x) แทนโหนดพ่อแม่ของโหนด $x(\neq r)$ ซึ่งเป็นโหนดประชิดของ x บนเส้นทาง T[x, r]

3.2 ความรู้พื้นฐานที่จำเป็นสำหรับขั้นตอนวิธี

สำหรับต้นไม้ T_i ใดๆ ที่มีโหนด r เป็นโหนดราก กำหนดให้ $x(\neq r)$ เป็นโหนดใดๆ บน T_i และ τ เป็นเซตของต้นไม้ทอดข้ามที่แตกต่าง k ต้นที่มีโหนดรากเป็นโหนด r บน Q_k ($\tau = \{T_i: 0 \leq i \leq k-1\}$) และจากคุณสมบัติการเชื่อมต่อของ Q_k และนิยามของ τ ทำให้โหนดราก (r) ของต้นไม้ทอดข้ามที่แตกต่าง T_i แต่ละต้นมีโหนดลูกเพียงโหนดเดียวเท่านั้นคือ $r+2^i$ ในกรณีที่ $r_i = 0$ หรือ $r-2^i$ ในกรณีที่ $r_i = 1$

ตัวอย่างของ T_0, T_1, T_2 และ T_3 ซึ่งมีโหนดรากเป็นโหนด 8 (r = 8) ของ Q_4 แสดงดังรูปที่ 3.1



รูปที่ 3.1 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ที่มีโหนดรากเป็นโหนด 8 (r=8)

สำหรับ Q_k ใดๆ ซึ่งประกอบด้วยโหนด r (โหนดต้นทาง) เป็นโหนดรากของต้นไม้ทอดข้ามที่แตกต่างและโหนด $x(\neq r)$ โดยหมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ $r = r_{k-1}r_{k-2}\ldots r_0$ และ $x = x_{k-1}x_{k-2}\ldots x_0$ จากการที่เลขฐานสองของ x และ r แตกต่างกัน ทำให้สามารถสร้างเซต $H_{x,r} = \{i: 0 \leq i \leq k-1 \text{ และ } x_i \neq r_i\}$ ซึ่งเป็นเซตของตำแหน่งของบิตต่าง (Difference bit) ระหว่าง x กับ r ได้ และจะสังเกตได้ว่าระยะทางแฮมมิ่ง (Hamming distance) ของทั้ง x และ r คือ $|H_{x,r}|$ นั่นเอง ยกตัวอย่างเช่น กำหนดให้โหนด r = 1000(8) และ x = 1111(15) เป็นสมาชิกของ Q_4 จะเห็นได้ว่าเลขฐานสองของทั้ง 2 โหนดแตกต่างกันตรงตำแหน่งที่ 0, 1 และ 2 ดังนั้นจะได้ $H_{15,8} = \{0, 1, 2\}$ และระยะทางแฮมมิ่งระหว่างโหนด 8 และ 15 = $|H_{15,8}| = 3$

จัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไป (Generalized Hamming Distance Latin Square: GHDLS) ซึ่งปรับปรุงมาจากจัตุรัสลาตินของระยะทางแฮมมิ่ง (Hamming Distance Latin Square: HDLS) [13] คือ เมตริกซ์ที่สมาชิกในแถวแรกทั้งหมดเป็นสมาชิกของ $H_{x,r}$ สมมติให้โหนด $x(\neq r)$ ใดๆ เป็นสมาชิกของ Q_k มีระยะทางแฮมมิ่งเมื่อเทียบกับโหนด r เท่ากับ t จะทำให้ได้ $H_{x,r} = \{i_0, i_1, \ldots, i_j, \ldots, i_{t-2}, i_{t-1}\}$ โดยที่ $i_0 < i_1 < \ldots < i_j < \ldots < i_{t-2} < i_{t-1}$ และสามารถสร้างจัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไปที่มีการหมุนรอบแบบเพิ่ม (Increasing GHDLS) ซึ่งเกี่ยวข้องกับ x และ r บน Q_k โดยใช้สัญลักษณ์แทนเป็น $I_k(x,r)$ ซึ่งแสดงเป็นเมตริกซ์ดังต่อไปนี้

$$I_k(x,r) = \begin{bmatrix} i_0 & i_1 & \cdots & i_{t-2} & i_{t-1} \\ i_1 & i_2 & \cdots & i_{t-1} & i_0 \\ \vdots & & \cdots & & \\ i_{t-2} & i_{t-1} & \cdots & i_{t-4} & i_{t-3} \\ i_{t-1} & i_0 & \cdots & i_{t-3} & i_{t-2} \end{bmatrix}$$

พิจารณาสมาชิกในเมตริกซ์ $I_k(x,r)$ สมาชิกตัวที่ j+1 หรือ i_{j+1} จะเรียกว่า ลูก (Successor) ของ i_j แสดงโดยใช้สัญลักษณ์ $succ(i_j)$ และสมาชิกตัวที่ j-1 หรือ i_{j-1} ใน $I_k(x,r)$ จะเรียกว่า พ่อแม่ (Predecessor) ของ i_j แสดงโดยใช้สัญลักษณ์ $pred(i_j)$ เมื่อพิจารณาบน Q_4 สมมติให้ r = 1000(8) เป็นโหนดราก และ x = 1111(15) จะสามารถสร้างเมตริกซ์ $I_4(15,8)$ ได้ดังตัวอย่างที่ 3.1 และหากกำหนดให้ x = 0111(7) จะสามารถสร้างเมตริกซ์ $I_4(7,8)$ ได้ดังตัวอย่างที่ 3.2

ตัวอย่างที่ 3.1 พิจารณาบน Q_4 โดยที่ r = 1000(8) เป็นโหนดราก และ x = 1111(15) จะได้ $H_{15,8} = \{0, 1, 2\}$ และจะทำให้สร้าง $I_4(15,8)$ ได้ดังเมตริกซ์ต่อไปนี้

$$I_4(15,8) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

จากเมตริกซ์ที่ได้หากพิจารณา $i_j = 2$ จะพบว่าลูกของ 2 หรือ succ(2) = 0 และพ่อแม่ของ 2 หรือ pred(2) = 1

ตัวอย่างที่ 3.2 พิจารณาเช่นเดียวกับตัวอย่างที่ 3.1 r = 1000(8) และ x = 0111(7) จะได้ $H_{7,8} = \{0, 1, 2, 3\}$ และจะทำให้สร้าง $I_4(7,8)$ ได้ดังเมตริกซ์ต่อไปนี้

$$I_4(7,8) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$$

จากเมตริกซ์ที่ได้หากพิจารณา $i_j = 2$ จะพบว่าลูกของ 2 หรือ succ(2) = 3 และพ่อแม่ของ 2 หรือ pred(2) = 1

3.3 ขั้นตอนวิธีสำหรับสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนดใดๆ บนเครือข่ายแบบไฮเปอร์คิว

จากหลักการของจัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไป ทำให้สามารถสร้างต้นไม้ทอดข้ามที่แตกต่าง k ต้นที่มีโหนดรากเป็นโหนดใดๆ บน Q_k ได้ด้วยขั้นตอนวิธีแบบขนานด้วยความซับซ้อนด้านเวลาเท่ากับ O(k) ในหัวข้อนี้จะนำเสนอขั้นตอนวิธีแบบขนานสำหรับสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนดใดๆ (r = 0, 1, 2, ..., หรือ 2^k-1) บนเครือข่ายแบบไฮเปอร์คิว ซึ่งขั้นตอนวิธีดังกล่าวแสดงได้ดังต่อไปนี้

```
Algorithm PARALLEL GEN-PARENTs-in-ISTs
begin
   for all vertex x(≠r) ∈ Qk pardo
      with r = rk-1rk-2...r0, x = xk-1xk-2...x0
      for i = 0 to k-1 do
         if (ri = xi)  // adjust bit weight at xi
               if (xi = 0) parent(Ti,x) = x + 2^i
               if (xi = 1) parent(Ti,x) = x - 2^i
         if (ri ≠ xi)  // apply GHDLS to adjust bit weight at xsucc(i)
               if (xsucc(i) = 0) parent(Ti,x) = x + 2^succ(i)
               if (xsucc(i) = 1) parent(Ti,x) = x - 2^succ(i)
      end for i
   end for all
end PARALLEL GEN-PARENTs-in-ISTs
```

จากขั้นตอนวิธีกำหนดให้หมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ $r = r_{k-1}r_{k-2}\ldots r_0$ และ $x = x_{k-1}x_{k-2}\ldots x_0$ ดังนั้นการหาเส้นทางที่แตกต่างกันในแต่ละ T_i ทำได้โดยให้ทุกๆ โหนด $x(\neq r)$ คำนวณหาโหนดพ่อแม่ (Parent determination) ในแต่ละ T_i $(0 \leq i \leq k-1)$ ตั้งแต่ต้นที่ T_0 ถึงต้นที่ T_{k-1} โดยทำการเปรียบเทียบหมายเลขประจำโหนดในรูปแบบเลขฐานสองของทั้ง r และ x จากบิตที่ 0 ถึงบิตที่ k-1 ซึ่งสามารถจำแนกวิธีการคำนวณหาโหนดพ่อแม่ได้เป็น 2 กรณีดังนี้

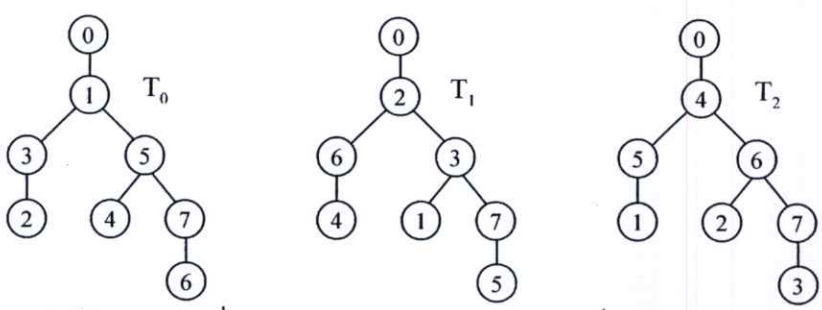
<u>กรณีที่ 1 : $(r_i = x_i)$</u> หรือกรณีที่ x เป็นโหนดใบไม้ (Leaf node) ในกรณีนี้สามารถคำนวณหาโหนดพ่อแม่ได้ด้วยวิธีการปรับบิตที่ i ของ x จาก 1 เป็น 0 หรือจาก 0 เป็น 1

<u>กรณีที่ 2 : $(r_i \neq x_i)$</u> หรือกรณีที่ x เป็นโหนดภายใน (Internal node) สามารถคำนวณหาโหนดพ่อแม่ได้โดยการปรับบิตของ x ตรงตำแหน่งของบิตต่างตำแหน่งถัดไป (succ(i)) จาก 1 เป็น 0 หรือจาก 0 เป็น 1 ซึ่งตำแหน่งของบิตต่างตำแหน่งถัดไปสามารถหาได้โดยการประยุกต์ใช้จัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไปของ x และ r หรือ $I_k(x,r)$

ตัวอย่างการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนดใดๆ r = 0, 1, 2, 3, 4, 5, 6, 7 บน Q_3 แสดงดังตัวอย่างที่ 3.3 ถึงตัวอย่างที่ 3.10 ตามลำดับ ในตัวอย่างที่ 3.3 จะทำการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีที่นำเสนอซึ่งจะเหมือนกับขั้นตอนวิธีของ J.-S. Yang และคณะ [13] ส่วนตัวอย่างที่ 3.4 ถึง 3.10 จะทำการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีที่นำเสนอ คือแบบทั่วไปในการสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนดใดๆ โดยหมายเลขประจำโหนดในรูปแบบเลขฐานสองของ $x = x_2x_1x_0$ และหมายเลขประจำโหนดในรูปแบบเลขฐานสองของ $r = r_2r_1r_0$

และสำหรับ Q_4 จะแสดงตัวอย่าง การสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ที่มีโหนดรากเป็นโหนดใดๆ เช่น r = 8 และ r = 13 ในตัวอย่างที่ 3.11 และ 3.12

<u>ตัวอย่างที่ 3.3</u> วิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 0 บน Q_3 ซึ่งการสร้างต้นไม้ทอดข้ามที่แตกต่างจะเหมือนกับขั้นตอนวิธีของ J.-S. Yang และคณะ โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 0 บน Q_3 แสดงดังรูปที่ 3.2



รูปที่ 3.2 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 0 (r = 0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 0)$ **ใน** T_0 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_0 = 0 (= r_0)$ จะได้ $x \in \{2, 4, 6\}$

ดังนั้น $parent(T_0, x) = x + 2^0 = x + 1$ จะได้ $parent(T_0, x) = \{3, 5, 7\}$

<u>กรณีที่ 2</u> ถ้า $x_0 = 1 (\neq r_0)$ จะแบ่งได้หลายกรณีดังนี้

$x \in \{1\}$ ดังนั้น $succ(0) = 0$ จะได้ $parent(T_0, x) = x - 2^0 = 1 - 1 = 0$

$x \in \{3, 7\}$ ดังนั้น $succ(0) = 1$ จะได้ $parent(T_0, x) = x - 2^1 = x - 2$

ดังนั้น $parent(T_0, x) = \{1, 5\}$

$x \in \{5\}$ ดังนั้น $succ(0) = 2$ จะได้ $parent(T_0, x) = x - 2^2 = 5 - 4 = 1$

ตารางที่ 3.1 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 0, i = 0)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
1	00***1***	0	1	1	00***0*** (0)
2	01***0***	-	-	-	01***1*** (3)
3	0***1***1	1	1	2	0***0***1 (1)
4	10***0***	-	-	-	10***1*** (5)
5	***1***01	2	1	4	***0***01 (1)
6	11***0***	-	-	-	11***1*** (7)
7	1***1***1	1	1	2	1***0***1 (5)

$I_3(1,0) = [\textcircled{0}]$ $\quad I_3(3,0) = \begin{bmatrix} 0 & \textcircled{1} \\ 1 & 0 \end{bmatrix}$ $\quad I_3(5,0) = \begin{bmatrix} 0 & \textcircled{2} \\ 2 & 0 \end{bmatrix}$ $\quad I_3(7,0) = \begin{bmatrix} 0 & \textcircled{1} & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 0)$ **ใน** T_1 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_1 = 0 (= r_1)$ จะได้ $x \in \{1, 4, 5\}$

ดังนั้น $parent(T_1, x) = x + 2^1 = x + 2$ จะได้ $parent(T_1, x) = \{3, 6, 7\}$

<u>กรณีที่ 2</u> ถ้า $x_1 = 1 (\neq r_1)$ จะแบ่งได้หลายกรณีดังนี้

$x \in \{3\}$ ดังนั้น $succ(1) = 0$ จะได้ $parent(T_1, x) = x - 2^0 = 3 - 1 = 2$

$x \in \{2\}$ ดังนั้น $succ(1) = 1$ จะได้ $parent(T_1, x) = x - 2^1 = 2 - 2 = 0$

$x \in \{6, 7\}$ ดังนั้น $succ(1) = 2$ จะได้ $parent(T_1, x) = x - 2^2$

ดังนั้น $parent(T_1, x) = \{2, 3\}$

ตารางที่ 3.2 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 0, i = 1)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
1	0*0*1	-	-	-	0*1*1 (3)
2	0*1*0	1	1	2	0*0*0 (0)
3	0*1*1	0	1	1	0*1*0 (2)
4	1*0*0	-	-	-	1*1*0 (6)
5	1*0*1	-	-	-	1*1*1 (7)
6	*1*10	2	1	4	*0*10 (2)
7	*1*11	2	1	4	*0*11 (3)

$I_3(2,0) = [①]$ $\quad I_3(3,0) = \begin{bmatrix} 0 & 1 \\ 1 & ⓪ \end{bmatrix}$ $\quad I_3(6,0) = \begin{bmatrix} 1 & ② \\ 2 & 1 \end{bmatrix}$ $\quad I_3(7,0) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & ② & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 0)$ **ใน** T_2 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_2 = 0 (= r_2)$ จะได้ $x \in \{1, 2, 3\}$

ดังนั้น $parent(T_2, x) = x + 2^2 = x + 4$ จะได้ $parent(T_2, x) = \{5, 6, 7\}$

<u>กรณีที่ 2</u> ถ้า $x_2 = 1 (\neq r_2)$ จะแบ่งได้หลายกรณีดังนี้

$x \in \{5, 7\}$ ดังนั้น $succ(2) = 0$ จะได้ $parent(T_2, x) = x - 2^0$

ดังนั้น $parent(T_2, x) = \{4, 6\}$

$x \in \{6\}$ ดังนั้น $succ(2) = 1$ จะได้ $parent(T_2, x) = x - 2^1 = 6 - 2 = 4$

$x \in \{4\}$ ดังนั้น $succ(2) = 2$ จะได้ $parent(T_2, x) = x - 2^2 = 4 - 4 = 0$

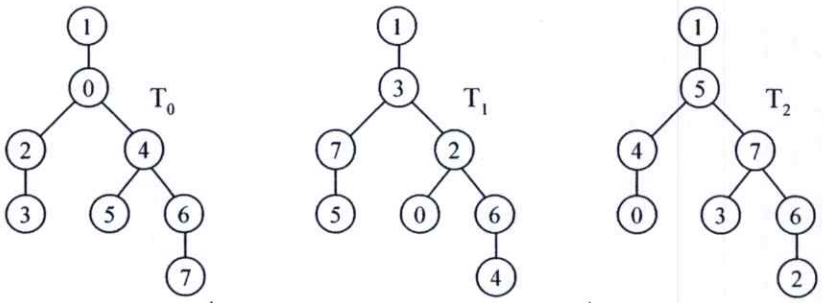
ตารางที่ 3.3 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 0, i = 2)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
1	*0*01	-	-	-	*1*01 (5)
2	*0*10	-	-	-	*1*10 (6)
3	*0*11	-	-	-	*1*11 (7)
4	*1*00	2	1	4	*0*00 (0)
5	10*1*	0	1	1	10*0* (4)
6	1*1*0	1	1	2	1*0*0 (4)
7	11*1*	0	1	1	11*0* (6)

$I_3(4,0) = [②]$ $\quad I_3(5,0) = \begin{bmatrix} 0 & 2 \\ 2 & ⓪ \end{bmatrix}$ $\quad I_3(6,0) = \begin{bmatrix} 1 & 2 \\ 2 & ① \end{bmatrix}$ $\quad I_3(7,0) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & ⓪ & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

ตัวอย่างที่ 3.4 วิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 001(1) บน Q_3 โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 1 บน Q_3 แสดงดังรูปที่ 3.3



รูปที่ 3.3 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 1 (r = 1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 1)$ **ใน** T_0 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_0 = r_0 = 1$ จะได้ $x \in \{3, 5, 7\}$

ดังนั้น $parent(T_0, x) = x - 2^0 = x - 1$ จะได้ $parent(T_0, x) = \{2, 4, 6\}$

<u>กรณีที่ 2</u> ถ้า $x_0 \neq r_0$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{0\}$ จะได้ $succ(0) = 0$ และ $x_{succ(0)} = x_0 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^0 = 0 + 1 = 1$

2. $x \in \{2, 6\}$ จะได้ $succ(0) = 1$ และ $x_{succ(0)} = x_1 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^1$ $\therefore$ $parent(T_0, x) = \{0, 4\}$

3. $x \in \{4\}$ จะได้ $succ(0) = 2$ และ $x_{succ(0)} = x_2 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^2 = 4 - 4 = 0$

ตารางที่ 3.4 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 1, i = 0)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	00***0***	0	0	1	00***1*** (1)
2	0***1***0	1	1	2	0***0***0 (0)
3	0***1***1	-	-	-	0***1***0 (2)
4	***1***00	2	1	4	***0***00 (0)
5	10***1***	-	-	-	10***0*** (4)
6	1***1***0	1	1	2	1***0***0 (4)
7	11***1***	-	-	-	11***0*** (6)

$I_3(0,1) = [\textcircled{0}]$ $\quad I_3(2,1) = \begin{bmatrix} 0 & \textcircled{1} \\ 1 & 0 \end{bmatrix}$ $\quad I_3(4,1) = \begin{bmatrix} 0 & \textcircled{2} \\ 2 & 0 \end{bmatrix}$ $\quad I_3(6,1) = \begin{bmatrix} 0 & \textcircled{1} & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 1)$ ใน T_1 บน Q_3

<u>กรณีที่ 1</u> ถ้า $x_1 = r_1 = 0$ จะได้ $x \in \{0, 4, 5\}$

ดังนั้น $parent(T_1, x) = x + 2^1 = x + 2$ จะได้ $parent(T_1, x) = \{2, 6, 7\}$

<u>กรณีที่ 2</u> ถ้า $x_1 \neq r_1$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{2\}$ จะได้ $succ(1) = 0$ และ $x_{succ(1)} = x_0 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^0 = 2 + 1 = 3$

2. $x \in \{3\}$ จะได้ $succ(1) = 1$ และ $x_{succ(1)} = x_1 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^1 = 3 - 2 = 1$

3. $x \in \{6, 7\}$ จะได้ $succ(1) = 2$ และ $x_{succ(1)} = x_2 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^2$ $\therefore$ $parent(T_1, x) = \{2, 3\}$

ตารางที่ 3.5 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 ($r = 1, i = 1$)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	0**0**0	-	-	-	0**1**0 (2)
2	0**1**0	0	0	1	0**1**1 (3)
3	0**1**1	1	1	2	0**0**1 (1)
4	1**0**0	-	-	-	1**1**0 (6)
5	1**0**1	-	-	-	1**1**1 (7)
6	**1**10	2	1	4	**0**10 (2)
7	**1**11	2	1	4	**0**11 (3)

$I_3(3,1) = [①]$ $\quad I_3(2,1) = \begin{bmatrix} 0 & 1 \\ 1 & ⓪ \end{bmatrix}$ $\quad I_3(7,1) = \begin{bmatrix} 1 & ② \\ 2 & 1 \end{bmatrix}$ $\quad I_3(6,1) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & ② & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 1)$ ใน T_2 บน Q_3

<u>กรณีที่ 1</u> ถ้า $x_2 = r_2 = 0$ จะได้ $x \in \{0, 2, 3\}$

ดังนั้น $parent(T_2, x) = x + 2^2 = x + 4$ จะได้ $parent(T_2, x) = \{4, 6, 7\}$

<u>กรณีที่ 2</u> ถ้า $x_2 \neq r_2$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{4, 6\}$ จะได้ $succ(2) = 0$ และ $x_{succ(2)} = x_0 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^0$ $\therefore$ $parent(T_2, x) = \{5, 7\}$

2. $x \in \{7\}$ จะได้ $succ(2) = 1$ และ $x_{succ(2)} = x_1 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^1 = 7 - 2 = 5$

3. $x \in \{5\}$ จะได้ $succ(2) = 2$ และ $x_{succ(2)} = x_2 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^2 = 5 - 4 = 1$

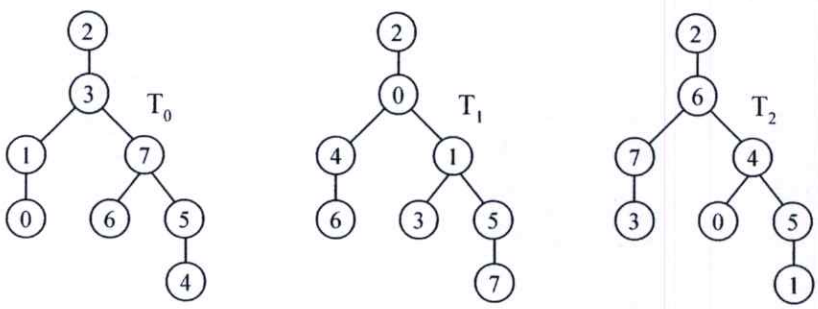
ตารางที่ 3.6 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 1, i = 2)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	***0***00	-	-	-	***1***00 (4)
2	***0***10	-	-	-	***1***10 (6)
3	***0***11	-	-	-	***1***11 (7)
4	10***0***	0	0	1	10***1*** (5)
5	***1***01	2	1	4	***0***01 (1)
6	11***0***	0	0	1	11***1*** (7)
7	1***1***1	1	1	2	1***0***1 (5)

$I_3(5,1) = [②]$ $\quad I_3(4,1) = \begin{bmatrix} 0 & 2 \\ 2 & ⓪ \end{bmatrix}$ $\quad I_3(7,1) = \begin{bmatrix} 1 & 2 \\ 2 & ① \end{bmatrix}$ $\quad I_3(6,1) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & ⓪ & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

ตัวอย่างที่ 3.5 วิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 010(2) บน Q_3 โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 2 บน Q_3 แสดงดังรูปที่ 3.4



รูปที่ 3.4 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 2 (r = 2)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 2)$ **ใน** T_0 **บน** Q_3

กรณีที่ 1 ถ้า $x_0 = r_0 = 0$ จะได้ $x \in \{0, 4, 6\}$

ดังนั้น $parent(T_0, x) = x + 2^0 = x + 1$ จะได้ $parent(T_0, x) = \{1, 5, 7\}$

กรณีที่ 2 ถ้า $x_0 \neq r_0$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{3\}$ จะได้ $succ(0) = 0$ และ $x_{succ(0)} = x_0 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^0 = 3 - 1 = 2$

2. $x \in \{1, 5\}$ จะได้ $succ(0) = 1$ และ $x_{succ(0)} = x_1 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^1$ $\therefore$ $parent(T_0, x) = \{3, 7\}$

3. $x \in \{7\}$ จะได้ $succ(0) = 2$ และ $x_{succ(0)} = x_2 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^2 = 7 - 4 = 3$

ตารางที่ 3.7 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 2, i = 0)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	00***0***	-	-	-	00***1*** (1)
1	0***0***1	1	0	2	0***1***1 (3)
3	01***1***	0	1	1	01***0*** (2)
4	10***0***	-	-	-	10***1*** (5)
5	1***0***1	1	0	2	1***1***1 (7)
6	11***0***	-	-	-	11***1*** (7)
7	***1***11	2	1	4	***0***11 (3)

$I_3(3,2) = [⓪]$ $\quad I_3(1,2) = \begin{bmatrix} 0 & ① \\ 1 & 0 \end{bmatrix}$ $\quad I_3(7,2) = \begin{bmatrix} 0 & ② \\ 2 & 0 \end{bmatrix}$ $\quad I_3(5,2) = \begin{bmatrix} 0 & ① & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 2)$ **ใน** T_1 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_1 = r_1 = 1$ จะได้ $x \in \{3, 6, 7\}$

ดังนั้น $parent(T_1, x) = x - 2^1 = x - 2$ จะได้ $parent(T_1, x) = \{1, 4, 5\}$

<u>กรณีที่ 2</u> ถ้า $x_1 \neq r_1$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{1\}$ จะได้ $succ(1) = 0$ และ $x_{succ(1)} = x_0 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^0 = 1 - 1 = 0$

2. $x \in \{0\}$ จะได้ $succ(1) = 1$ และ $x_{succ(1)} = x_1 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^1 = 0 + 2 = 2$

3. $x \in \{4, 5\}$ จะได้ $succ(1) = 2$ และ $x_{succ(1)} = x_2 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^2$ $\therefore$ $parent(T_1, x) = \{0, 1\}$

ตารางที่ 3.8 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 2, i = 1)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	0***0***0	1	0	2	0***1***0 (2)
1	00***1***	0	1	1	00***0*** (0)
3	0***1***1	-	-	-	0***0***1 (1)
4	***1***00	2	1	4	***0***00 (0)
5	***1***01	2	1	4	***0***01 (1)
6	1***1***0	-	-	-	1***0***0 (4)
7	1***1***1	-	-	-	1***0***1 (5)

$I_3(0,2) = [①]$ $\quad I_3(1,2) = \begin{bmatrix} 0 & 1 \\ 1 & ⓪ \end{bmatrix}$ $\quad I_3(4,2) = \begin{bmatrix} 1 & ② \\ 2 & 1 \end{bmatrix}$ $\quad I_3(5,2) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & ② & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 2)$ **ใน** T_2 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_2 = r_2 = 0$ จะได้ $x \in \{0, 1, 3\}$

ดังนั้น $parent(T_2, x) = x + 2^2 = x + 4$ จะได้ $parent(T_2, x) = \{4, 5, 7\}$

<u>กรณีที่ 2</u> ถ้า $x_2 \neq r_2$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{5, 7\}$ จะได้ $succ(2) = 0$ และ $x_{succ(2)} = x_0 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^0 \therefore parent(T_2, x) = \{4, 6\}$

2. $x \in \{4\}$ จะได้ $succ(2) = 1$ และ $x_{succ(2)} = x_1 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^1 = 4 + 2 = 6$

3. $x \in \{6\}$ จะได้ $succ(2) = 2$ และ $x_{succ(2)} = x_2 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^2 = 6 - 4 = 2$

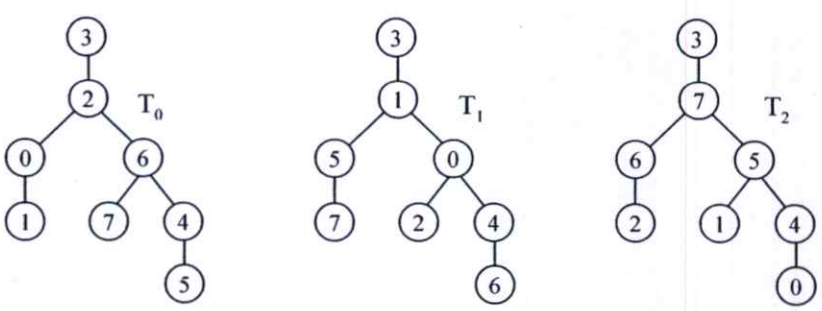
ตารางที่ 3.9 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 ($r = 2, i = 2$)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	***0***00	-	-	-	***1***00 (4)
1	***0***01	-	-	-	***1***01 (5)
3	***0***11	-	-	-	***1***11 (7)
4	1***0***0	1	0	2	1***1***0 (6)
5	10***1***	0	1	1	10***0*** (4)
6	***1***10	2	1	4	***0***10 (2)
7	11***1***	0	1	1	11***0*** (6)

$I_3(6,2) = [②]$ $\quad I_3(4,2) = \begin{bmatrix} 1 & 2 \\ 2 & ① \end{bmatrix}$ $\quad I_3(7,2) = \begin{bmatrix} 0 & 2 \\ 2 & ⓪ \end{bmatrix}$ $\quad I_3(5,2) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & ⓪ & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

<u>ตัวอย่างที่ 3.6</u> วิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 011(3) บน Q_3 โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 3 บน Q_3 แสดงดังรูปที่ 3.5



รูปที่ 3.5 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 3 ($r = 3$)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 3)$ **ใน** T_0 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_0 = r_0 = 1$ จะได้ $x \in \{1, 5, 7\}$

ดังนั้น $parent(T_0, x) = x - 2^0 = x - 1$ จะได้ $parent(T_0, x) = \{0, 4, 6\}$

<u>กรณีที่ 2</u> ถ้า $x_0 \neq r_0$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{2\}$ จะได้ $succ(0) = 0$ และ $x_{succ(0)} = x_0 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^0 = 2 + 1 = 3$

2. $x \in \{0, 4\}$ จะได้ $succ(0) = 1$ และ $x_{succ(0)} = x_1 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^1 \therefore parent(T_0, x) = \{2, 6\}$

3. $x \in \{6\}$ จะได้ $succ(0) = 2$ และ $x_{succ(0)} = x_2 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^2 = 6 - 4 = 2$

ตารางที่ 3.10 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 ($r = 3, i = 0$)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	0***0***0	1	0	2	0***1***0 (2)
1	0***0***1	-	-	-	0***0***0 (0)
2	0***1***0	0	0	1	0***1***1 (3)
4	1***0***0	1	0	2	1***1***0 (6)
5	1***0***1	-	-	-	1***0***0 (4)
6	***1***10	2	1	4	***0***10 (2)
7	11***1***	-	-	-	11***0*** (6)

$I_3(2,3) = [\textcircled{0}]$ $\quad I_3(0,3) = \begin{bmatrix} 0 & \textcircled{1} \\ 1 & 0 \end{bmatrix}$ $\quad I_3(6,3) = \begin{bmatrix} 0 & \textcircled{2} \\ 2 & 0 \end{bmatrix}$ $\quad I_3(4,3) = \begin{bmatrix} 0 & \textcircled{1} & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 3)$ **ใน** T_1 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_1 = r_1 = 1$ จะได้ $x \in \{2, 6, 7\}$

ดังนั้น $parent(T_1, x) = x - 2^1 = x - 2$ จะได้ $parent(T_1, x) = \{0, 4, 5\}$

<u>กรณีที่ 2</u> ถ้า $x_1 \neq r_1$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{0\}$ จะได้ $succ(1) = 0$ และ $x_{succ(1)} = x_0 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^0 = 0 + 1 = 1$

2. $x \in \{1\}$ จะได้ $succ(1) = 1$ และ $x_{succ(1)} = x_1 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^1 = 1 + 2 = 3$

3. $x \in \{4, 5\}$ จะได้ $succ(1) = 2$ และ $x_{succ(1)} = x_2 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^2 \therefore parent(T_1, x) = \{0, 1\}$

ตารางที่ 3.11 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 3, i = 1)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	00***0***	0	0	1	00***1*** (1)
1	0***0***1	1	0	2	0***1***1 (3)
2	0***1***0	-	-	-	0***0***0 (0)
4	***1***00	2	1	4	***0***00 (0)
5	***1***01	2	1	4	***0***01 (1)
6	1***1***0	-	-	-	1***0***0 (4)
7	1***1***1	-	-	-	1***0***1 (5)

$I_3(1,3) = [①]$ $\quad I_3(0,3) = \begin{bmatrix} 0 & 1 \\ 1 & ⓪ \end{bmatrix}$ $\quad I_3(5,3) = \begin{bmatrix} 1 & ② \\ 2 & 1 \end{bmatrix}$ $\quad I_3(4,3) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & ② & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 3)$ **ใน** T_2 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_2 = r_2 = 0$ จะได้ $x \in \{0, 1, 2\}$

ดังนั้น $parent(T_2, x) = x + 2^2 = x + 4$ จะได้ $parent(T_2, x) = \{4, 5, 6\}$

<u>กรณีที่ 2</u> ถ้า $x_2 \neq r_2$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{4, 6\}$ จะได้ $succ(2) = 0$ และ $x_{succ(2)} = x_0 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^0$ $\therefore$ $parent(T_2, x) = \{5, 7\}$

2. $x \in \{5\}$ จะได้ $succ(2) = 1$ และ $x_{succ(2)} = x_1 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^1 = 5 + 2 = 7$

3. $x \in \{7\}$ จะได้ $succ(2) = 2$ และ $x_{succ(2)} = x_2 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^2 = 7 - 4 = 3$

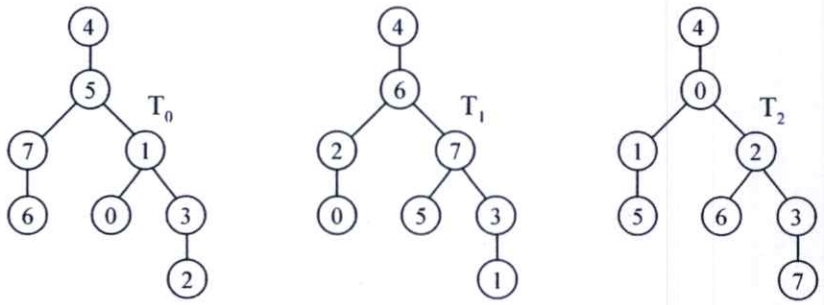
ตารางที่ 3.12 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 3, i = 2)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	***0***00	-	-	-	***1***00 (4)
1	***0***01	-	-	-	***1***01 (5)
2	***0***10	-	-	-	***1***10 (6)
4	10***0***	0	0	1	10***1*** (5)
5	1***0***1	1	0	2	1***1***1 (7)
6	11***0***	0	0	1	11***1*** (7)
7	***1***11	2	1	4	***0***11 (3)

$I_3(7,3) = [②]$ $\quad I_3(5,3) = \begin{bmatrix} 1 & 2 \\ 2 & ① \end{bmatrix}$ $\quad I_3(6,3) = \begin{bmatrix} 0 & 2 \\ 2 & ⓪ \end{bmatrix}$ $\quad I_3(4,3) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & ⓪ & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

ตัวอย่างที่ 3.7 วิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 100(4) บน Q_3 โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 4 บน Q_3 แสดงดังรูปที่ 3.6



รูปที่ 3.6 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 4 (r = 4)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 4)$ **ใน** T_0 **บน** Q_3

กรณีที่ 1 ถ้า $x_0 = r_0 = 0$ จะได้ $x \in \{0, 2, 6\}$

ดังนั้น $parent(T_0, x) = x + 2^0 = x + 1$ จะได้ $parent(T_0, x) = \{1, 3, 7\}$

กรณีที่ 2 ถ้า $x_0 \neq r_0$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{5\}$ จะได้ $succ(0) = 0$ และ $x_{succ(0)} = x_0 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^0 = 5 - 1 = 4$

2. $x \in \{3, 7\}$ จะได้ $succ(0) = 1$ และ $x_{succ(0)} = x_1 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^1 \therefore parent(T_0, x) = \{1, 5\}$

3. $x \in \{1\}$ จะได้ $succ(0) = 2$ และ $x_{succ(0)} = x_2 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^2 = 1 + 4 = 5$

ตารางที่ 3.13 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 4, i = 0)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	00***0***	-	-	-	00***1*** (1)
1	***0***01	2	0	4	***1***01 (5)
2	01***0***	-	-	-	01***1*** (3)
3	0***1***1	1	1	2	0***0***1 (1)
5	10***1***	0	1	1	10***0*** (4)
6	11***0***	-	-	-	11***1*** (7)
7	1***1***1	1	1	2	1***0***1 (5)

$I_3(5,4) = [\textcircled{0}]$ $\quad I_3(1,4) = \begin{bmatrix} 0 & \textcircled{2} \\ 2 & 0 \end{bmatrix}$ $\quad I_3(7,4) = \begin{bmatrix} 0 & \textcircled{1} \\ 1 & 0 \end{bmatrix}$ $\quad I_3(3,4) = \begin{bmatrix} 0 & \textcircled{1} & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 4)$ **ใน** T_1 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_1 = r_1 = 0$ จะได้ $x \in \{0, 1, 5\}$

ดังนั้น $parent(T_1, x) = x + 2^1 = x + 2$ จะได้ $parent(T_1, x) = \{2, 3, 7\}$

<u>กรณีที่ 2</u> ถ้า $x_1 \neq r_1$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{7\}$ จะได้ $succ(1) = 0$ และ $x_{succ(1)} = x_0 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^0 = 7 - 1 = 6$

2. $x \in \{6\}$ จะได้ $succ(1) = 1$ และ $x_{succ(1)} = x_1 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^1 = 6 - 2 = 4$

3. $x \in \{2, 3\}$ จะได้ $succ(1) = 2$ และ $x_{succ(1)} = x_2 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^2$ $\therefore$ $parent(T_1, x) = \{6, 7\}$

ตารางที่ 3.14 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 ($r = 4$, $i = 1$)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	0***0***0	-	-	-	0***1***0 (2)
1	0***0***1	-	-	-	0***1***1 (3)
2	***0***10	2	0	4	***1***10 (6)
3	***0***11	2	0	4	***1***11 (7)
5	1***0***1	-	-	-	1***1***1 (7)
6	1***1***0	1	1	2	1***0***0 (4)
7	11***1***	0	1	1	11***0*** (6)

$I_3(6,4) = [①]$ $\quad I_3(2,4) = \begin{bmatrix} 1 & ② \\ 2 & 1 \end{bmatrix}$ $\quad I_3(7,4) = \begin{bmatrix} 0 & 1 \\ 1 & ⓪ \end{bmatrix}$ $\quad I_3(3,4) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & ② & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 4)$ **ใน** T_2 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_2 = r_2 = 1$ จะได้ $x \in \{5, 6, 7\}$

ดังนั้น $parent(T_2, x) = x - 2^2 = x - 4$ จะได้ $parent(T_2, x) = \{1, 2, 3\}$

<u>กรณีที่ 2</u> ถ้า $x_2 \neq r_2$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{1, 3\}$ จะได้ $succ(2) = 0$ และ $x_{succ(2)} = x_0 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^0$ $\therefore$ $parent(T_2, x) = \{0, 2\}$

2. $x \in \{2\}$ จะได้ $succ(2) = 1$ และ $x_{succ(2)} = x_1 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^1 = 2 - 2 = 0$

3. $x \in \{0\}$ จะได้ $succ(2) = 2$ และ $x_{succ(2)} = x_2 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^2 = 0 + 4 = 4$

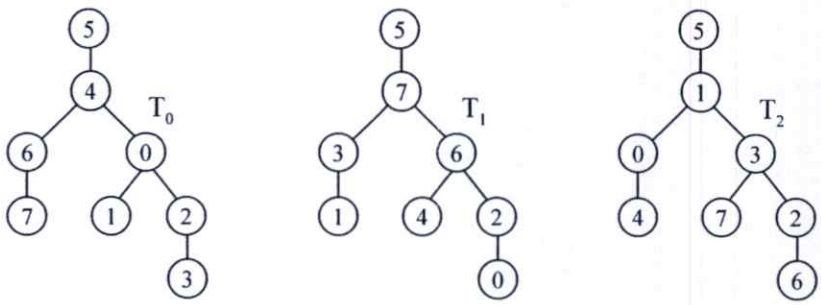
ตารางที่ 3.15 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 (r = 4, i = 2)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	***0***00	2	0	4	***1***00 (4)
1	00***1***	0	1	1	00***0*** (0)
2	0***1***0	1	1	2	0***0***0 (0)
3	01***1***	0	1	1	01***0*** (2)
5	***1***01	-	-	-	***0***01 (1)
6	***1***10	-	-	-	***0***10 (2)
7	***1***11	-	-	-	***0***11 (3)

$I_3(0,4) = [②]$ $\quad I_3(1,4) = \begin{bmatrix} 0 & 2 \\ 2 & ⓪ \end{bmatrix}$ $\quad I_3(2,4) = \begin{bmatrix} 1 & 2 \\ 2 & ① \end{bmatrix}$ $\quad I_3(3,4) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & ⓪ & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

ตัวอย่างที่ 3.8 วิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 101(5) บน Q_3 โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 5 บน Q_3 แสดงดังรูปที่ 3.7



รูปที่ 3.7 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 5 (r = 5)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 5)$ **ใน** T_0 **บน** Q_3

กรณีที่ 1 ถ้า $x_0 = r_0 = 1$ จะได้ $x \in \{1, 3, 7\}$

ดังนั้น $parent(T_0, x) = x - 2^0 = x - 1$ จะได้ $parent(T_0, x) = \{0, 2, 6\}$

กรณีที่ 2 ถ้า $x_0 \neq r_0$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{4\}$ จะได้ $succ(0) = 0$ และ $x_{succ(0)} = x_0 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^0 = 4 + 1 = 5$

2. $x \in \{2, 6\}$ จะได้ $succ(0) = 1$ และ $x_{succ(0)} = x_1 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^1$ $\therefore$ $parent(T_0, x) = \{0, 4\}$

3. $x \in \{0\}$ จะได้ $succ(0) = 2$ และ $x_{succ(0)} = x_2 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^2 = 0 + 4 = 4$

ตารางที่ 3.16 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 5, i = 0)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	***0***00	2	0	4	***1***00 (4)
1	00***1***	-	-	-	00***0*** (0)
2	0***1***0	1	1	2	0***0***0 (0)
3	01***1***	-	-	-	01***0*** (2)
4	10***0***	0	0	1	10***1*** (5)
6	1***1***0	1	1	2	1***0***0 (4)
7	11***1***	-	-	-	11***0*** (6)

$I_3(4,5) = [⓪]$ $\quad I_3(0,5) = \begin{bmatrix} 0 & ② \\ 2 & 0 \end{bmatrix}$ $\quad I_3(6,5) = \begin{bmatrix} 0 & ① \\ 1 & 0 \end{bmatrix}$ $\quad I_3(2,5) = \begin{bmatrix} 0 & ① & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 5)$ **ใน** T_1 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_1 = r_1 = 0$ จะได้ $x \in \{0, 1, 4\}$

ดังนั้น $parent(T_1, x) = x + 2^1 = x + 2$ จะได้ $parent(T_1, x) = \{2, 3, 6\}$

<u>กรณีที่ 2</u> ถ้า $x_1 \neq r_1$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{6\}$ จะได้ $succ(1) = 0$ และ $x_{succ(1)} = x_0 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^0 = 6 + 1 = 7$

2. $x \in \{7\}$ จะได้ $succ(1) = 1$ และ $x_{succ(1)} = x_1 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^1 = 7 - 2 = 5$

3. $x \in \{2, 3\}$ จะได้ $succ(1) = 2$ และ $x_{succ(1)} = x_2 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^2$ $\therefore$ $parent(T_1, x) = \{6, 7\}$

ตารางที่ 3.17 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 (r = 5, i = 1)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	0***0***0	-	-	-	0***1***0 (2)
1	0***0***1	-	-	-	0***1***1 (3)
2	***0***10	2	0	4	***1***10 (6)
3	***0***11	2	0	4	***1***11 (7)
4	1***0***0	-	-	-	1***1***0 (6)
6	11***0***	0	0	1	11***1*** (7)
7	1***1***1	1	1	2	1***0***1 (5)

$I_3(7,5) = [①]$ $\quad I_3(3,5) = \begin{bmatrix} 1 & ② \\ 2 & 1 \end{bmatrix}$ $\quad I_3(6,5) = \begin{bmatrix} 0 & 1 \\ 1 & ⓪ \end{bmatrix}$ $\quad I_3(2,5) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & ② & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 5)$ **ใน** T_2 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_2 = r_2 = 1$ จะได้ $x \in \{4, 6, 7\}$

ดังนั้น $parent(T_2, x) = x - 2^2 = x - 4$ จะได้ $parent(T_2, x) = \{0, 2, 3\}$

<u>กรณีที่ 2</u> ถ้า $x_2 \neq r_2$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{0, 2\}$ จะได้ $succ(2) = 0$ และ $x_{succ(2)} = x_0 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^0$ $\therefore$ $parent(T_2, x) = \{1, 3\}$

2. $x \in \{3\}$ จะได้ $succ(2) = 1$ และ $x_{succ(2)} = x_1 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^1 = 3 - 2 = 1$

3. $x \in \{1\}$ จะได้ $succ(2) = 2$ และ $x_{succ(2)} = x_2 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^2 = 1 + 4 = 5$

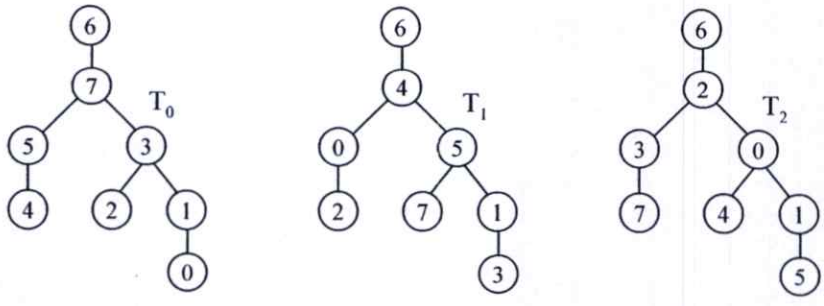
ตารางที่ 3.18 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 ($r = 5$, $i = 2$)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	00***0***	0	0	1	00***1*** (1)
1	***0***01	2	0	4	***1***01 (5)
2	01***0***	0	0	1	01***1*** (3)
3	0***1***1	1	1	2	0***0***1 (1)
4	***1***00	-	-	-	***0***00 (0)
6	***1***10	-	-	-	***0***10 (2)
7	***1***11	-	-	-	***0***11 (3)

$I_3(1,5) = [②]$ $\quad I_3(0,5) = \begin{bmatrix} 0 & 2 \\ 2 & ⓪ \end{bmatrix}$ $\quad I_3(3,5) = \begin{bmatrix} 1 & 2 \\ 2 & ① \end{bmatrix}$ $\quad I_3(2,5) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & ⓪ & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

<u>ตัวอย่างที่ 3.9</u> วิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 110(6) บน Q_3 โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 6 บน Q_3 แสดงดังรูปที่ 3.8



รูปที่ 3.8 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 6 ($r = 6$)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 6)$ **ใน** T_0 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_0 = r_0 = 0$ จะได้ $x \in \{0, 2, 4\}$

ดังนั้น $parent(T_0, x) = x + 2^0 = x + 1$ จะได้ $parent(T_0, x) = \{1, 3, 5\}$

<u>กรณีที่ 2</u> ถ้า $x_0 \neq r_0$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{7\}$ จะได้ $succ(0) = 0$ และ $x_{succ(0)} = x_0 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^0 = 7 - 1 = 6$

2. $x \in \{1, 5\}$ จะได้ $succ(0) = 1$ และ $x_{succ(0)} = x_1 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^1$ $\therefore$ $parent(T_0, x) = \{3, 7\}$

3. $x \in \{3\}$ จะได้ $succ(0) = 2$ และ $x_{succ(0)} = x_2 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^2 = 3 + 4 = 7$

ตารางที่ 3.19 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 ($r = 6$, $i = 0$)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	00***0***	-	-	-	00***1*** (1)
1	0***0***1	1	0	2	0***1***1 (3)
2	01***0***	-	-	-	01***1*** (3)
3	***0***11	2	0	4	***1***11 (7)
4	10***0***	-	-	-	10***1*** (5)
5	1***0***1	1	0	2	1***1***1 (7)
7	11***1***	0	1	1	11***0*** (6)

$I_3(7,6) = [\textcircled{0}]$ $\quad I_3(3,6) = \begin{bmatrix} 0 & \textcircled{2} \\ 2 & 0 \end{bmatrix}$ $\quad I_3(5,6) = \begin{bmatrix} 0 & \textcircled{1} \\ 1 & 0 \end{bmatrix}$ $\quad I_3(1,6) = \begin{bmatrix} 0 & \textcircled{1} & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 6)$ **ใน** T_1 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_1 = r_1 = 1$ จะได้ $x \in \{2, 3, 7\}$

ดังนั้น $parent(T_1, x) = x - 2^1 = x - 2$ จะได้ $parent(T_1, x) = \{0, 1, 5\}$

<u>กรณีที่ 2</u> ถ้า $x_1 \neq r_1$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{5\}$ จะได้ $succ(1) = 0$ และ $x_{succ(1)} = x_0 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^0 = 5 - 1 = 4$

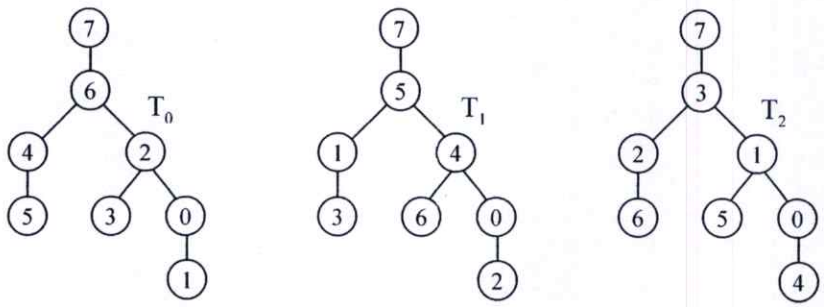
2. $x \in \{4\}$ จะได้ $succ(1) = 1$ และ $x_{succ(1)} = x_1 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^1 = 4 + 2 = 6$

3. $x \in \{0, 1\}$ จะได้ $succ(1) = 2$ และ $x_{succ(1)} = x_2 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^2$ $\therefore$ $parent(T_1, x) = \{4, 5\}$

ตัวอย่างที่ 3.10 วิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 111(7) บน Q_3 โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ที่มีโหนดรากเป็นโหนด 7 บน Q_3 แสดงดังรูปที่ 3.9



รูปที่ 3.9 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1 และ T_2 ของ Q_3 ที่มีโหนดรากเป็นโหนด 7 (r = 7)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 7)$ **ใน** T_0 **บน** Q_3

กรณีที่ 1 ถ้า $x_0 = r_0 = 1$ จะได้ $x \in \{1, 3, 5\}$

ดังนั้น $parent(T_0, x) = x - 2^0 = x - 1$ จะได้ $parent(T_0, x) = \{0, 2, 4\}$

กรณีที่ 2 ถ้า $x_0 \neq r_0$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{6\}$ จะได้ $succ(0) = 0$ และ $x_{succ(0)} = x_0 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^0 = 6 + 1 = 7$

2. $x \in \{0, 4\}$ จะได้ $succ(0) = 1$ และ $x_{succ(0)} = x_1 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^1 \quad \therefore \ parent(T_0, x) = \{2, 6\}$

3. $x \in \{2\}$ จะได้ $succ(0) = 2$ และ $x_{succ(0)} = x_2 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^2 = 2 + 4 = 6$

ตารางที่ 3.22 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_3 (r = 7, i = 0)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	0**0**0	1	0	2	0**1**0 (2)
1	00**1**	-	-	-	00**0** (0)
2	**0**10	2	0	4	**1**10 (6)
3	01**1**	-	-	-	01**0** (2)
4	1**0**0	1	0	2	1**1**0 (6)
5	10**1**	-	-	-	10**0** (4)
6	11**0**	0	0	1	11**1** (7)

$I_3(6,7) = [\textcircled{0}]$ $\quad I_3(2,7) = \begin{bmatrix} 0 & \textcircled{2} \\ 2 & 0 \end{bmatrix}$ $\quad I_3(4,7) = \begin{bmatrix} 0 & \textcircled{1} \\ 1 & 0 \end{bmatrix}$ $\quad I_3(0,7) = \begin{bmatrix} 0 & \textcircled{1} & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 7)$ **ใน** T_1 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_1 = r_1 = 1$ จะได้ $x \in \{2, 3, 6\}$

ดังนั้น $parent(T_1, x) = x - 2^1 = x - 2$ จะได้ $parent(T_1, x) = \{0, 1, 4\}$

<u>กรณีที่ 2</u> ถ้า $x_1 \neq r_1$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{4\}$ จะได้ $succ(1) = 0$ และ $x_{succ(1)} = x_0 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^0 = 4 + 1 = 5$

2. $x \in \{5\}$ จะได้ $succ(1) = 1$ และ $x_{succ(1)} = x_1 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^1 = 5 + 2 = 7$

3. $x \in \{0, 1\}$ จะได้ $succ(1) = 2$ และ $x_{succ(1)} = x_2 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^2$ $\therefore$ $parent(T_1, x) = \{4, 5\}$

ตารางที่ 3.23 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_3 ($r = 7$, $i = 1$)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	***0***00	2	0	4	***1***00 (4)
1	***0***01	2	0	4	***1***01 (5)
2	0***1***0	-	-	-	0***0***0 (0)
3	0***1***1	-	-	-	0***0***1 (1)
4	10***0***	0	0	1	10***1*** (5)
5	1***0***1	1	0	2	1***1***1 (7)
6	1***1***0	-	-	-	1***0***0 (4)

$I_3(5,7) = [①]$ $\quad I_3(1,7) = \begin{bmatrix} 1 & ② \\ 2 & 1 \end{bmatrix}$ $\quad I_3(4,7) = \begin{bmatrix} 0 & 1 \\ 1 & ⓪ \end{bmatrix}$ $\quad I_3(0,7) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & ② & 0 \\ 2 & 0 & 1 \end{bmatrix}$

เมื่อ ○ แสดงค่าของ succ(1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 7)$ **ใน** T_2 **บน** Q_3

<u>กรณีที่ 1</u> ถ้า $x_2 = r_2 = 1$ จะได้ $x \in \{4, 5, 6\}$

ดังนั้น $parent(T_2, x) = x - 2^2 = x - 4$ จะได้ $parent(T_2, x) = \{0, 1, 2\}$

<u>กรณีที่ 2</u> ถ้า $x_2 \neq r_2$ จะแบ่งได้ 3 กรณีดังนี้

1. $x \in \{0, 2\}$ จะได้ $succ(2) = 0$ และ $x_{succ(2)} = x_0 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^0$ $\therefore$ $parent(T_2, x) = \{1, 3\}$

2. $x \in \{1\}$ จะได้ $succ(2) = 1$ และ $x_{succ(2)} = x_1 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^1 = 1 + 2 = 3$

3. $x \in \{3\}$ จะได้ $succ(2) = 2$ และ $x_{succ(2)} = x_2 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^2 = 3 + 4 = 7$

ตารางที่ 3.24 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_3 ($r = 7, i = 2$)

x	$x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	00***0***	0	0	1	00***1*** (1)
1	0***0***1	1	0	2	0***1***1 (3)
2	01***0***	0	0	1	01***1*** (3)
3	***0***11	2	0	4	***1***11 (7)
4	***1***00	-	-	-	***0***00 (0)
5	***1***01	-	-	-	***0***01 (1)
6	***1***10	-	-	-	***0***10 (2)

$I_3(3,7) = [\textcircled{2}]$ $\quad I_3(1,7) = \begin{bmatrix} 1 & 2 \\ 2 & \textcircled{1} \end{bmatrix}$ $\quad I_3(2,7) = \begin{bmatrix} 0 & 2 \\ 2 & \textcircled{0} \end{bmatrix}$ $\quad I_3(0,7) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & \textcircled{0} & 1 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

ตัวอย่างที่ 3.11 แสดงวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ที่มีโหนด 8 เป็นโหนดราก ($r = 8$) ในที่นี้หมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ $x = x_3x_2x_1x_0$ และหมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ $r = r_3r_2r_1r_0$ โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ซึ่งมีโหนด 8 เป็นโหนดรากแสดงดังรูปที่ 3.1 และข้อมูลทั้งหมดสำหรับการสร้าง T_0, T_1, T_2 และ T_3 แสดงดังตารางที่ 3.25, 3.26, 3.27 และ 3.28 ตามลำดับ

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 8)$ **ใน** T_0 **บน** Q_4

กรณีที่ 1 ถ้า $x_0 = r_0 = 0$ จะได้ $x \in \{0, 2, 4, 6, 10, 12, 14\}$

ดังนั้น $parent(T_0, x) = x + 2^0 = x + 1$ จะได้ $parent(T_0, x) = \{1, 3, 5, 7, 11, 13, 15\}$

กรณีที่ 2 ถ้า $x_0 \neq r_0$ จะแบ่งได้ 4 กรณีดังนี้

1. $x \in \{9\}$ จะได้ $succ(0) = 0$ และ $x_{succ(0)} = x_0 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^0 = 9 - 1 = 8$

2. $x \in \{3, 7, 11, 15\}$ จะได้ $succ(0) = 1$ และ $x_{succ(0)} = x_1 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^1$ $\therefore$ $parent(T_0, x) = \{1, 5, 9, 13\}$

3. $x \in \{5, 13\}$ จะได้ $succ(0) = 2$ และ $x_{succ(0)} = x_2 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^2$ $\therefore$ $parent(T_0, x) = \{1, 9\}$

4. $x \in \{1\}$ จะได้ $succ(0) = 3$ และ $x_{succ(0)} = x_3 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^3 = 1 + 8 = 9$

ตารางที่ 3.25 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_4 (r = 8, i = 0)

x	$x_3x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	000***0***	-	-	-	000***1*** (1)
1	***0***001	3	0	8	***1***001 (9)
2	001***0***	-	-	-	001***1*** (3)
3	00***11***	1	1	2	00***0***1 (1)
4	010***0***	-	-	-	010***1*** (5)
5	0***1***01	2	1	4	0***0***01 (1)
6	011***0***	-	-	-	011***1*** (7)
7	01***11***	1	1	2	01***0***1 (5)
9	100***1***	0	1	1	100***0*** (8)
10	101***0***	-	-	-	101***1*** (11)
11	10***11***	1	1	2	10***0***1 (9)
12	110***0***	-	-	-	110***1*** (13)
13	1***1***01	2	1	4	1***0***01 (9)
14	111***0***	-	-	-	111***1*** (15)
15	11***11***	1	1	2	11***0***1 (13)

$l_4(9,8) = [\bigcirc 0]$

$l_4(1,8) = \begin{bmatrix} 0 & \bigcirc 3 \\ 3 & 0 \end{bmatrix}$

$l_4(11,8) = \begin{bmatrix} 0 & \bigcirc 1 \\ 1 & 0 \end{bmatrix}$

$l_4(13,8) = \begin{bmatrix} 0 & \bigcirc 2 \\ 2 & 0 \end{bmatrix}$

$l_4(3,8) = \begin{bmatrix} 0 & \bigcirc 1 & 3 \\ 1 & 3 & 0 \\ 3 & 0 & 1 \end{bmatrix}$

$l_4(5,8) = \begin{bmatrix} 0 & \bigcirc 2 & 3 \\ 2 & 3 & 0 \\ 3 & 0 & 2 \end{bmatrix}$

$l_4(15,8) = \begin{bmatrix} 0 & \bigcirc 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

$l_4(7,8) = \begin{bmatrix} 0 & \bigcirc 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

เมื่อ ○ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 8)$ **ใน** T_1 **บน** Q_4

<u>กรณีที่ 1</u> ถ้า $x_1 = r_1 = 0$ จะได้ $x \in \{0, 1, 4, 5, 9, 12, 13\}$

ดังนั้น $parent(T_1, x) = x + 2^1 = x + 2$ จะได้ $parent(T_1, x) = \{2, 3, 6, 7, 11, 14, 15\}$

<u>กรณีที่ 2</u> ถ้า $x_1 \neq r_1$ จะแบ่งได้ 4 กรณีดังนี้

1. $x \in \{11\}$ จะได้ $succ(1) = 0$ และ $x_{succ(1)} = x_0 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^0 = 11 - 1 = 10$

2. $x \in \{10\}$ จะได้ $succ(1) = 1$ และ $x_{succ(1)} = x_1 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^1 = 10 - 2 = 8$

3. $x \in \{6, 7, 14, 15\}$ จะได้ $succ(1) = 2$ และ $x_{succ(1)} = x_2 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^2$ $\therefore$ $parent(T_1, x) = \{2, 3, 10, 11\}$

4. $x \in \{2, 3\}$ จะได้ $succ(1) = 3$ และ $x_{succ(1)} = x_3 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^3$ $\therefore$ $parent(T_1, x) = \{10, 11\}$

ตารางที่ 3.26 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_4 (r = 8, i = 1)

x	$x_3x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	00***0***	-	-	-	00***1***0 (2)
1	00***0***1	-	-	-	00***1***1 (3)
2	***0***010	3	0	8	***1***010 (10)
3	***0***011	3	0	8	***1***011 (11)
4	01***0***0	-	-	-	01***1***0 (6)
5	01***0***1	-	-	-	01***1***1 (7)
6	0***1***10	2	1	4	0***0***10 (2)
7	0***1***11	2	1	4	0***0***11 (3)
9	10***0***1	-	-	-	10***1***1 (11)
10	10***1***0	1	1	2	100***0*** (8)
11	101***1***	0	1	1	101***0*** (10)
12	11***0***0	-	-	-	11***1***0 (14)
13	11***0***1	-	-	-	11***1***1 (15)
14	1***1***10	2	1	4	1***0***10 (10)
15	1***1***11	2	1	4	1***0***11 (11)

$I_4(10,8) = [\,\textcircled{1}\,]$

$I_4(2,8) = \begin{bmatrix} 1 & \textcircled{3} \\ 3 & 1 \end{bmatrix}$

$I_4(11,8) = \begin{bmatrix} 0 & 1 \\ 1 & \textcircled{0} \end{bmatrix}$

$I_4(14,8) = \begin{bmatrix} 1 & \textcircled{2} \\ 2 & 1 \end{bmatrix}$

$I_4(3,8) = \begin{bmatrix} 0 & 1 & 3 \\ 1 & \textcircled{3} & 0 \\ 3 & 0 & 1 \end{bmatrix}$

$I_4(6,8) = \begin{bmatrix} 1 & \textcircled{2} & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$

เมื่อ ○ แสดงค่าของ succ(1)

$I_4(15,8) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & \textcircled{2} & 0 \\ 2 & 0 & 1 \end{bmatrix}$

$I_4(7,8) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & \textcircled{2} & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 8)$ **ใน** T_2 **บน** Q_4

<u>กรณีที่ 1</u> ถ้า $x_2 = r_2 = 0$ จะได้ $x \in \{0, 1, 2, 3, 9, 10, 11\}$

ดังนั้น $parent(T_2, x) = x + 2^2 = x + 4$ จะได้ $parent(T_2, x) = \{4, 5, 6, 7, 13, 14, 15\}$

<u>กรณีที่ 2</u> ถ้า $x_2 \neq r_2$ จะแบ่งได้ 4 กรณีดังนี้

1. $x \in \{13, 15\}$ จะได้ $succ(2) = 0$ และ $x_{succ(2)} = x_0 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^0$ $\therefore$ $parent(T_2, x) = \{12, 14\}$

2. $x \in \{14\}$ จะได้ $succ(2) = 1$ และ $x_{succ(2)} = x_1 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^1 = 14 - 2 = 12$

3. $x \in \{12\}$ จะได้ $succ(2) = 2$ และ $x_{succ(2)} = x_2 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^2 = 12 - 4 = 8$

4. $x \in \{4, 5, 6, 7\}$ จะได้ $succ(2) = 3$ และ $x_{succ(2)} = x_3 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^3$ $\therefore$ $parent(T_2, x) = \{12, 13, 14, 15\}$

ตารางที่ 3.27 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_4 (r = 8, i = 2)

x	$x_3x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	0***0***00	-	-	-	0***1***00 (4)
1	0***0***01	-	-	-	0***1***01 (5)
2	0***0***10	-	-	-	0***1***10 (6)
3	0***0***11	-	-	-	0***1***11 (7)
4	***0***100	3	0	8	***1***100 (12)
5	***0***101	3	0	8	***1***101 (13)
6	***0***110	3	0	8	***1***110 (14)
7	***0***111	3	0	8	***1***111 (15)
9	1***0***01	-	-	-	1***1***01 (13)
10	1***0***10	-	-	-	1***1***10 (14)
11	1***0***11	-	-	-	1***1***11 (15)
12	1***1***00	2	1	4	1***0***00 (8)
13	1101	0	1	1	1100 (12)
14	111***0***	1	1	2	110***0*** (12)
15	1111	0	1	1	111***0*** (14)

$I_4(12,8) = [\,(2)\,]$

$I_4(4,8) = \begin{bmatrix} 2 & (3) \\ 3 & 2 \end{bmatrix}$

$I_4(13,8) = \begin{bmatrix} 0 & 2 \\ 2 & (0) \end{bmatrix}$

$I_4(14,8) = \begin{bmatrix} 1 & 2 \\ 2 & (1) \end{bmatrix}$

$I_4(5,8) = \begin{bmatrix} 0 & 2 & 3 \\ 2 & (3) & 0 \\ 3 & 0 & 2 \end{bmatrix}$

$I_4(6,8) = \begin{bmatrix} 1 & 2 & 3 \\ 2 & (3) & 1 \\ 3 & 1 & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

$I_4(15,8) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & (0) & 1 \end{bmatrix}$

$I_4(7,8) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & (3) & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 8)$ **ใน** T_3 **บน** Q_4

<u>กรณีที่ 1</u> ถ้า $x_3 = r_3 = 1$ จะได้ $x \in \{9, 10, 11, 12, 13, 14, 15\}$

ดังนั้น $parent(T_3, x) = x - 2^3 = x - 8$ จะได้ $parent(T_3, x) = \{1, 2, 3, 4, 5, 6, 7\}$

<u>กรณีที่ 2</u> ถ้า $x_3 \neq r_3$ จะแบ่งได้ 4 กรณีดังนี้

1. $x \in \{1, 3, 5, 7\}$ จะได้ $succ(3) = 0$ และ $x_{succ(3)} = x_0 = 1$

ดังนั้น $parent(T_3, x) = x - 2^{succ(3)} = x - 2^0 \quad \therefore \; parent(T_3, x) = \{0, 2, 4, 6\}$

2. $x \in \{2, 6\}$ จะได้ $succ(3) = 1$ และ $x_{succ(3)} = x_1 = 1$

ดังนั้น $parent(T_3, x) = x - 2^{succ(3)} = x - 2^1 \quad \therefore \; parent(T_3, x) = \{0, 4\}$

3. $x \in \{4\}$ จะได้ $succ(3) = 2$ และ $x_{succ(3)} = x_2 = 1$

ดังนั้น $parent(T_3, x) = x - 2^{succ(3)} = x - 2^2 = 4 - 4 = 0$

4. $x \in \{0\}$ จะได้ $succ(3) = 3$ และ $x_{succ(3)} = x_3 = 0$

ดังนั้น $parent(T_3, x) = x + 2^{succ(3)} = x + 2^3 = 0 + 8 = 8$

ตารางที่ 3.28 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_3 บน Q_4 (r = 8, i = 3)

x	$x_3x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	***0***000	3	0	8	***1***000 (8)
1	000***1***	0	1	1	000***0*** (0)
2	00***1***0	1	1	2	00***0***0 (0)
3	001***1***	0	1	1	001***0*** (2)
4	0***1***00	2	1	4	0***0***00 (0)
5	010***1***	0	1	1	010***0*** (4)
6	01***1***0	1	1	2	01***0***0 (4)
7	011***1***	0	1	1	011***0*** (6)
9	***1***001	-	-	-	***0***001 (1)
10	***1***010	-	-	-	***0***010 (2)
11	***1***011	-	-	-	***0***011 (3)
12	***1***100	-	-	-	***0***100 (4)
13	***1***101	-	-	-	***0***101 (5)
14	***1***110	-	-	-	***0***110 (6)
15	***1***111	-	-	-	***0***111 (7)

$I_4(0,8) = [\,(3)\,]$

$I_4(1,8) = \begin{bmatrix} 0 & 3 \\ 3 & (0) \end{bmatrix}$

$I_4(2,8) = \begin{bmatrix} 1 & 3 \\ 3 & (1) \end{bmatrix}$

$I_4(4,8) = \begin{bmatrix} 2 & 3 \\ 3 & (2) \end{bmatrix}$

$I_4(3,8) = \begin{bmatrix} 0 & 1 & 3 \\ 1 & 3 & 0 \\ 3 & (0) & 1 \end{bmatrix}$

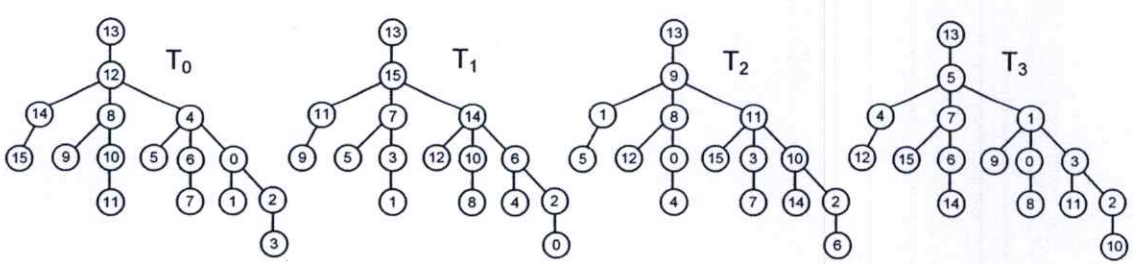
$I_4(5,8) = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 3 & 0 \\ 3 & (0) & 2 \end{bmatrix}$

$I_4(6,8) = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & (1) & 2 \end{bmatrix}$

$I_4(7,8) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & (0) & 1 & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(3)

<u>ตัวอย่างที่ 3.12</u> แสดงวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ซึ่งมีโหนด 13 เป็นโหนดราก (r = 13) ในที่นี้หมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ x = $x_3x_2x_1x_0$ และหมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ r = $r_3r_2r_1r_0$ โดยต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ซึ่งมีโหนด 13 เป็นโหนดรากแสดงดังรูปที่ 3.10



รูปที่ 3.10 แสดงต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ที่มีโหนดรากเป็นโหนด 13 (r = 13)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 13)$ **ใน** T_0 **บน** Q_4

<u>กรณีที่ 1</u> ถ้า $x_0 = r_0 = 1$ จะได้ $x \in \{1, 3, 5, 7, 9, 11, 15\}$

ดังนั้น $parent(T_0, x) = x - 2^0 = x - 1$ จะได้ $parent(T_0, x) = \{0, 2, 4, 6, 8, 10, 14\}$

<u>กรณีที่ 2</u> ถ้า $x_0 \neq r_0$ จะแบ่งได้ 4 กรณีดังนี้

1. $x \in \{12\}$ จะได้ $succ(0) = 0$ และ $x_{succ(0)} = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = 13$

2. $x \in \{2, 6, 10, 14\}$ จะได้ $succ(0) = 1$ และ $x_{succ(0)} = x_1 = 1$

ดังนั้น $parent(T_0, x) = x - 2^{succ(0)} = x - 2^1$ $\therefore$ $parent(T_0, x) = \{0, 4, 8, 12\}$

3. $x \in \{0, 8\}$ จะได้ $succ(0) = 2$ และ $x_{succ(0)} = x_2 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = x + 2^2$ $\therefore$ $parent(T_0, x) = \{4, 12\}$

4. $x \in \{4\}$ จะได้ $succ(0)=3$ และ $x_{succ(0)} = x_3 = 0$

ดังนั้น $parent(T_0, x) = x + 2^{succ(0)} = 12$

ตารางที่ 3.29 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_0 บน Q_4 (r = 13, i = 0)

x	$x_3x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	0***0***00	2	0	4	0***1***00 (4)
1	000***1***	-	-	-	000***0*** (0)
2	00***1***0	1	1	2	00***0***0 (0)
3	001***1***	-	-	-	001***0*** (2)
4	***0***100	3	0	8	***1***100 (12)
5	010***1***	-	-	-	010***0*** (4)
6	01***1***0	1	1	2	01***0***0 (4)
7	011***1***	-	-	-	011***0*** (6)
8	1***0***00	2	0	4	1***1***00 (12)
9	100***1***	-	-	-	100***0*** (8)
10	10***1***0	1	1	2	10***0***0 (8)
11	101***1***	-	-	-	101***0*** (10)
12	110***0***	0	0	1	110***1*** (13)
14	11***1***0	1	1	2	11***0***0 (12)
15	1111	-	-	-	111***0*** (14)

$I_4(12,13) = [0]$

$I_4(4,13) = \begin{bmatrix} 0 & 3 \\ 3 & 0 \end{bmatrix}$

$I_4(8,13) = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$

$I_4(14,13) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

$I_4(0,13) = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 3 & 0 \\ 3 & 0 & 2 \end{bmatrix}$

$I_4(6,13) = \begin{bmatrix} 0 & 1 & 3 \\ 1 & 3 & 0 \\ 3 & 0 & 1 \end{bmatrix}$

$I_4(10,13) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$

$I_4(2,13) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(0)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 13)$ ใน T_1 บน Q_4

<u>กรณีที่ 1</u> ถ้า $x_1 = r_1 = 0$ จะได้ $x \in \{0, 1, 4, 5, 8, 9, 12\}$

ดังนั้น $parent(T_1, x) = x + 2^1 = x + 2$ จะได้ $parent(T_1, x) = \{2, 3, 6, 7, 10, 11, 14\}$

<u>กรณีที่ 2</u> ถ้า $x_1 \neq r_1$ จะแบ่งได้ 4 กรณีดังนี้

1. $x \in \{14\}$ จะได้ $succ(1) = 0$ และ $x_{succ(1)} = x_0 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^0 = 14 + 1 = 15$

2. $x \in \{15\}$ จะได้ $succ(1) = 1$ และ $x_{succ(1)} = x_1 = 1$

ดังนั้น $parent(T_1, x) = x - 2^{succ(1)} = x - 2^1 = 15 - 2 = 13$

3. $x \in \{2, 3, 10, 11\}$ จะได้ $succ(1) = 2$ และ $x_{succ(1)} = x_2 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^2$ $\therefore$ $parent(T_1, x) = \{6, 7, 14, 15\}$

4. $x \in \{6, 7\}$ จะได้ $succ(1) = 3$ และ $x_{succ(1)} = x_3 = 0$

ดังนั้น $parent(T_1, x) = x + 2^{succ(1)} = x + 2^3$ $\therefore$ $parent(T_1, x) = \{14, 15\}$

ตารางที่ 3.30 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_1 บน Q_4 ($r = 13$, $i = 1$)

x	$x_3x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	parent(T_i, x)
0	00***0***	-	-	-	00***1***0 (2)
1	00***0***	-	-	-	00***1***1 (3)
2	0***0***10	2	0	4	0***1***10 (6)
3	0***0***11	2	0	4	0***1***11 (7)
4	01***0***0	-	-	-	01***1***0 (6)
5	01***0***1	-	-	-	01***1***1 (7)
6	***0***110	3	0	8	***1***110 (14)
7	***0***111	3	0	8	***1***111 (15)
8	10***0***0	-	-	-	10***1***0 (10)
9	10***0***1	-	-	-	10***1***1 (11)
10	1***0***10	2	0	4	1***1***10 (14)
11	1***0***11	2	0	4	1***1***11 (15)
12	11***0***0	-	-	-	11***1***0 (14)
14	111***0***	0	0	1	111***1*** (15)
15	11***1***1	1	1	2	11***0***1 (13)

$I_4(15,13) = [\,(1)\,]$

$I_4(7,13) = \begin{bmatrix} 1 & (3) \\ 3 & 1 \end{bmatrix}$

$I_4(11,13) = \begin{bmatrix} 1 & (2) \\ 2 & 1 \end{bmatrix}$

$I_4(14,13) = \begin{bmatrix} 0 & (1) \\ 1 & 0 \end{bmatrix}$

$I_4(3,13) = \begin{bmatrix} 1 & (2) & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$

$I_4(6,13) = \begin{bmatrix} 0 & 1 & 3 \\ 1 & (3) & 0 \\ 3 & 0 & 1 \end{bmatrix}$

$I_4(10,13) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & (2) & 0 \\ 2 & 0 & 1 \end{bmatrix}$

$I_4(2,13) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & (2) & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(1)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 13)$ **ใน** T_2 **บน** Q_4

<u>กรณีที่ 1</u> ถ้า $x_2 = r_2 = 1$ จะได้ $x \in \{4, 5, 6, 7, 12, 14, 15\}$

ดังนั้น $parent(T_2, x) = x - 2^2 = x - 4$ จะได้ $parent(T_2, x) = \{0, 1, 2, 3, 8, 10, 11\}$

<u>กรณีที่ 2</u> ถ้า $x_2 \neq r_2$ จะแบ่งได้ 4 กรณีดังนี้

1. $x \in \{8, 10\}$ จะได้ $succ(2) = 0$ และ $x_{succ(2)} = x_0 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^0$ $\therefore$ $parent(T_2, x) = \{9, 11\}$

2. $x \in \{11\}$ จะได้ $succ(2) = 1$ และ $x_{succ(2)} = x_1 = 1$

ดังนั้น $parent(T_2, x) = x - 2^{succ(2)} = x - 2^1 = 11 - 2 = 9$

3. $x \in \{9\}$ จะได้ $succ(2) = 2$ และ $x_{succ(2)} = x_2 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^2 = 9 + 4 = 13$

4. $x \in \{0, 1, 2, 3\}$ จะได้ $succ(2) = 3$ และ $x_{succ(2)} = x_3 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(2)} = x + 2^3$ $\therefore$ $parent(T_2, x) = \{8, 9, 10, 11\}$

ตารางที่ 3.31 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_2 บน Q_4 ($r = 13$, $i = 2$)

x	$x_3x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	***0***000	3	0	8	***1***000 (8)
1	***0***001	3	0	8	***1***001 (9)
2	***0***010	3	0	8	***1***010 (10)
3	***0***011	3	0	8	***1***011 (11)
4	0***1***00	-	-	-	0***0***00 (0)
5	0***1***01	-	-	-	0***0***01 (1)
6	0***1***10	-	-	-	0***0***10 (2)
7	0***1***11	-	-	-	0***0***11 (3)
8	100***0***	0	0	1	100***1*** (9)
9	1***0***01	2	0	4	1***1***01 (13)
10	101***0***	0	0	1	101***1*** (11)
11	10***1***1	1	1	2	10***0***1 (9)
12	1***1***00	-	-	-	1***0***00 (8)
14	1***1***10	-	-	-	1***0***10 (10)
15	1***1***11	-	-	-	1***0***11 (11)

$I_4(9,13) = [2]$

$I_4(1,13) = \begin{bmatrix} 2 & \textcircled{3} \\ 3 & 2 \end{bmatrix}$

$I_4(8,13) = \begin{bmatrix} 0 & 2 \\ 2 & \textcircled{0} \end{bmatrix}$

$I_4(11,13) = \begin{bmatrix} 1 & 2 \\ 2 & \textcircled{1} \end{bmatrix}$

$I_4(0,13) = \begin{bmatrix} 0 & 2 & 3 \\ 2 & \textcircled{3} & 0 \\ 3 & 0 & 2 \end{bmatrix}$

$I_4(3,13) = \begin{bmatrix} 1 & 2 & 3 \\ 2 & \textcircled{3} & 1 \\ 3 & 1 & 2 \end{bmatrix}$

$I_4(10,13) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & \textcircled{0} & 1 \end{bmatrix}$

$I_4(2,13) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & \textcircled{3} & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(2)

วิธีการคำนวณหาโหนดพ่อแม่ของโหนด $x(\neq 13)$ **ใน** T_3 **บน** Q_4

<u>กรณีที่ 1</u> ถ้า $x_3 = r_3 = 1$ จะได้ $x \in \{8, 9, 10, 11, 12, 14, 15\}$

ดังนั้น $parent(T_3, x) = x - 2^3 = x - 8$ จะได้ $parent(T_3, x) = \{0, 1, 2, 3, 4, 6, 7\}$

<u>กรณีที่ 2</u> ถ้า $x_3 \neq r_3$ จะแบ่งได้ 4 กรณีดังนี้

1. $x \in \{0, 2, 4, 6\}$ จะได้ $succ(3) = 0$ และ $x_{succ(3)} = x_0 = 0$

ดังนั้น $parent(T_3, x) = x + 2^{succ(3)} = x + 2^0$ $\therefore$ $parent(T_3, x) = \{1, 3, 5, 7\}$

2. $x \in \{3, 7\}$ จะได้ $succ(3) = 1$ และ $x_{succ(3)} = x_1 = 1$

ดังนั้น $parent(T_3, x) = x - 2^{succ(3)} = x - 2^1$ $\therefore$ $parent(T_3, x) = \{1, 5\}$

3. $x \in \{1\}$ จะได้ $succ(3) = 2$ และ $x_{succ(3)} = x_2 = 0$

ดังนั้น $parent(T_3, x) = x + 2^{succ(3)} = x + 2^2 = 1 + 4 = 5$

4. $x \in \{5\}$ จะได้ $succ(3) = 3$ และ $x_{succ(3)} = x_3 = 0$

ดังนั้น $parent(T_2, x) = x + 2^{succ(3)} = x + 2^3 = 5 + 8 = 13$

ตารางที่ 3.32 แสดงข้อมูลทั้งหมดสำหรับการสร้าง T_3 บน Q_4 ($r = 13$, $i = 3$)

x	$x_3x_2x_1x_0$	succ(i)	$x_{succ(i)}$	$2^{succ(i)}$	$parent(T_i, x)$
0	000***0***	0	0	1	000***1*** (1)
1	0***0***01	2	0	4	0***1***01 (5)
2	001***0***	0	0	1	001***1*** (3)
3	00***1***1	1	1	2	00***0***1 (1)
4	010***0***	0	0	1	010***1*** (5)
5	***0***101	3	0	8	***1***101 (13)
6	011***0***	0	0	1	011***1*** (7)
7	01***1***1	1	1	2	01***0***1 (5)
8	***1***000	-	-	-	***0***000 (0)
9	***1***001	-	-	-	***0***001 (1)
10	***1***010	-	-	-	***0***010 (2)
11	***1***011	-	-	-	***0***011 (3)
12	***1***100	-	-	-	***0***100 (4)
14	***1***110	-	-	-	***0***110 (6)
15	***1***111	-	-	-	***0***111 (7)

$I_4(5,13) = [\,(3)\,]$

$I_4(1,13) = \begin{bmatrix} 2 & 3 \\ 3 & (2) \end{bmatrix}$

$I_4(4,13) = \begin{bmatrix} 0 & 3 \\ 3 & (0) \end{bmatrix}$

$I_4(7,13) = \begin{bmatrix} 1 & 3 \\ 3 & (1) \end{bmatrix}$

$I_4(0,13) = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 3 & 0 \\ 3 & (0) & 2 \end{bmatrix}$

$I_4(3,13) = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & (1) & 2 \end{bmatrix}$

$I_4(6,13) = \begin{bmatrix} 0 & 1 & 3 \\ 1 & 3 & 0 \\ 3 & (0) & 1 \end{bmatrix}$

$I_4(2,13) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & (0) & 1 & 2 \end{bmatrix}$

เมื่อ ◯ แสดงค่าของ succ(3)

จากตัวอย่างข้างต้นจะพบว่าการสร้างต้นไม้ทอดข้ามที่แตกต่าง k ต้นที่มีโหนดรากเป็นโหนดใดๆ ($r = 0, 1, 2, \ldots,$ หรือ 2^k-1) ของ Q_k สามารถทำได้ในเวลา k รอบ โดยทำการเปรียบเทียบหมายเลขประจำโหนดในรูปแบบของเลขฐานสองของโหนด $x(\neq r)$ และโหนดราก r ตั้งแต่บิตที่ 0 ถึงบิตที่ k-1 นอกจากนี้หลักการของจัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไปที่นำมาประยุกต์ใช้ในขั้นตอนวิธียังสามารถทำให้เส้นทางจากโหนดใดๆ ไปยังโหนดรากในต้นไม้ทอดข้ามที่แตกต่างทั้ง k ต้นแตกต่างกันในทุกเส้นทางและทุกลิงค์ โดยการวิเคราะห์ความซับซ้อนด้านเวลาและความถูกต้องของขั้นตอนวิธีจะอธิบายในบทถัดไป

บทที่ 4
การวิเคราะห์ขั้นตอนวิธี

จากวิธีการแบบทั่วไปในการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีแบบขนานบนเครือข่ายแบบไฮเปอร์คิว พร้อมกับตัวอย่างบางส่วนที่ได้เสนอไว้ในบทที่ 3 ในบทนี้จะกล่าวถึงการวิเคราะห์ความซับซ้อนด้านเวลาและความถูกต้องของขั้นตอนวิธี ในส่วนของการวิเคราะห์ความถูกต้องของขั้นตอนวิธีจะทำการพิสูจน์ใน 3 ทฤษฎีบทด้วยกันได้แก่ ทฤษฎีบทที่ 1 จะเป็นการพิสูจน์ว่ามีต้นไม้ทอดข้ามที่แตกต่าง T_i ($0 \leq i \leq k-1$) ซึ่งมีโหนดรากเป็นโหนด r (r = 0, 1, 2, ..., หรือ 2^k-1) จำนวน k ต้นสำหรับเครือข่ายแบบไฮเปอร์คิว k มิติ และหากนับจำนวนเส้นเชื่อมบนเส้นทางเป็นระยะห่างระหว่างโหนด ทฤษฎีบทที่ 2 จะเป็นการพิสูจน์ว่าความสูงของต้นไม้ทอดข้ามที่แตกต่าง k ต้นที่มีโหนดรากเป็นโหนด r ใดๆ มีความสูงที่เหมาะสมคือ มีความสูงไม่เกิน k+1 เส้นเชื่อม และทฤษฎีบทที่ 3 จะพิสูจน์ว่าในต้นไม้ทอดข้ามที่แตกต่างใดๆ เส้นทางจากโหนด $x(\neq r)$ ซึ่งเป็นสมาชิกของเครือข่ายแบบไฮเปอร์คิว k มิติไปยังโหนด r เป็นเส้นทางที่แตกต่างกัน ในที่นี้จะพิจารณาหมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ $r = r_{k-1}r_{k-2}\ldots r_0$ และ $x = x_{k-1}x_{k-2}\ldots x_0$

4.1 การวิเคราะห์ความซับซ้อนด้านเวลา

เมื่อพิจารณาเครือข่ายแบบไฮเปอร์คิว k มิติใดๆ ในรูปแบบของกราฟ กราฟดังกล่าวจะประกอบด้วย $N = 2^k$ โหนด และในการประมวลผลแบบขนานบนเครือข่ายแบบไฮเปอร์คิว k มิติ ทุกๆ โหนด $x(\neq r)$ จะถูกกำหนดให้ทำการประมวลผลพร้อมกันด้วยคำสั่ง "for all vertex $x(\neq r) \in Q_k$ pardo" คำสั่งนี้มีความซับซ้อนด้านเวลาเท่ากับ O(1) และเมื่อพิจารณา หมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ $r = r_{k-1}r_{k-2}\ldots r_0$ และ $x = x_{k-1}x_{k-2}\ldots x_0$ แล้วจะสามารถวิเคราะห์ความซับซ้อนด้านเวลาของขั้นตอนการสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนดใดๆ (r = 0, 1, 2, ..., หรือ 2^k-1) ได้ดังนี้

```
for i = 0 to k-1 do
    if (ri = xi)  // adjust bit weight
        if (xi = 0) parent(Ti,x) = x + 2^i
        if (xi = 1) parent(Ti,x) = x – 2^i
    if (ri ≠ xi)  // apply GHDLS
        if (xsucc(i) = 0) parent(Ti,x) = x + 2^succ(i)
        if (xsucc(i) = 1) parent(Ti,x) = x – 2^succ(i)
end for i
```

ในขั้นตอนนี้ทุกๆ โหนด $x(\neq r)$ ซึ่งมีหมายเลขประจำโหนดในรูปแบบของเลขฐานสองเป็น $x = x_{k-1}x_{k-2}\ldots x_0$ จะทำการคำนวณหาโหนดพ่อแม่ในแต่ละ T_i $(0 \leq i \leq k-1)$ พร้อมกัน โดยทำการเปรียบเทียบบิต x_i กับ r_i ในแต่ละรอบ i เนื่องจากหมายเลขประจำโหนดในรูปแบบของเลขฐานสองของทั้ง x และ r มีจำนวน k บิต ดังนั้นการประมวลผลในขั้นตอนนี้จึงเท่ากับ k รอบ คือ ตั้งแต่รอบที่ i = 0 จนถึงรอบที่ i = k-1 และมีความซับซ้อนด้านเวลาเท่ากับ O(k) และเมื่อพิจารณาการประมวลผลในแต่ละรอบ i ซึ่งเป็นการเปรียบเทียบเงื่อนไขเพียง 2 เงื่อนไขเท่านั้นคือ

1. กรณีที่ $r_i = x_i$ จะทำการคำนวณหาโหนดพ่อแม่ของ x โดยการปรับบิต x_i ให้เป็นบิตตรงกันข้าม
2. กรณีที่ $r_i \neq x_i$ จะทำการพิจารณา GHDLS เพื่อหาตำแหน่งบิตต่างที่ถัดจาก i หรือ succ(i) และทำการคำนวณหาโหนดพ่อแม่ของ x โดยการปรับบิตที่ $x_{succ(i)}$ ให้เป็นบิตตรงกันข้าม

ซึ่งขั้นตอนนี้มีความซับซ้อนด้านเวลาเท่ากับ O(1)

จากการวิเคราะห์ข้างต้นจึงสรุปได้ว่าขั้นตอนการสร้างต้นไม้ทอดข้ามที่แตกต่างโดยการคำนวณหาโหนดพ่อแม่ของแต่ละโหนด $x(\neq r)$ ในแต่ละ T_i $(0 \leq i \leq k-1)$ ทั้ง k ต้น มีความซับซ้อนด้านเวลาเท่ากับ O(k)

4.2 การวิเคราะห์ความถูกต้องของขั้นตอนวิธี

ทฤษฎีบทที่ 1 สำหรับเครือข่ายแบบไฮเปอร์คิว k มิติใดๆ จะมีต้นไม้ทอดข้ามที่แตกต่าง T_i $(0 \leq i \leq k-1)$ ซึ่งมีโหนดรากเป็นโหนด r (r = 0, 1, 2, …, หรือ 2^k-1) จำนวน k ต้น

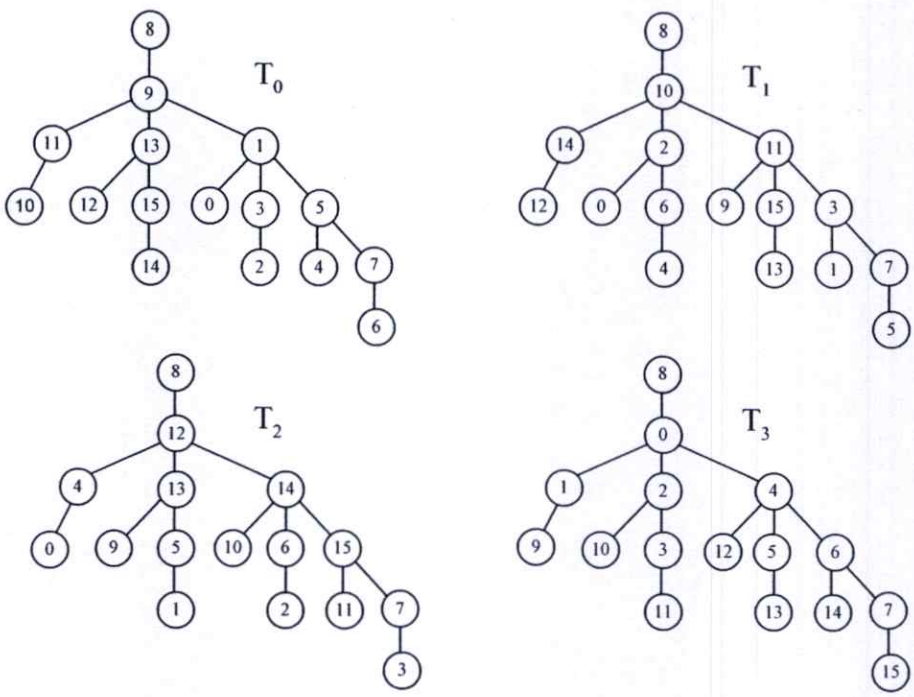
พิสูจน์ จากคุณสมบัติการเชื่อมต่อระหว่างโหนดของ Q_k ที่ว่า โหนดแต่ละโหนดบนไฮเปอร์คิวจะมีเส้นเชื่อมต่อกับโหนดประชิดในแต่ละมิติทั้งหมด k เส้นเชื่อม ดังนั้นโหนด r จึงมีโหนดประชิดทั้งหมด k โหนด และหากพิจารณาหมายเลขประจำโหนดในรูปแบบของเลขฐานสองจะพบว่าโหนด 2 โหนดใดๆ ที่อยู่ติดกันบนเส้นทางใน T_i จะมีระยะทางแฮมมิ่งเท่ากับ 1 และเมื่อพิจารณา $r = r_{k-1}r_{k-2} \ldots r_0$ จะเห็นได้ว่าหมายเลขประจำโหนดของโหนดลูกซึ่งอยู่ติดกับโหนด r สามารถคำนวณหาได้จากการปรับบิตของ r ทั้ง k บิตตั้งแต่บิต r_0, r_1 จนถึง r_{k-1} เพื่อให้ได้หมายเลขประจำโหนดของโหนดลูกของโหนด r ในต้นไม้ทอดข้ามที่แตกต่าง k ต้น ตั้งแต่ T_0, T_1 จนถึง T_{k-1} ตามลำดับ โดยการคำนวณหาหมายเลขประจำโหนดของโหนดลูกของโหนด r พิจารณาได้ใน 2 กรณีดังนี้

กรณีที่ $r_i = 0$ คำนวณหาหมายเลขประจำโหนดของโหนดลูกโดยปรับบิตที่ i จาก 0 เป็น 1 $(r + 2^i)$

กรณีที่ $r_i = 1$ คำนวณหาหมายเลขประจำโหนดของโหนดลูกโดยปรับบิตที่ i จาก 1 เป็น 0 $(r - 2^i)$

จากความสัมพันธ์ของโหนด r และโหนดลูกของโหนด r บนแต่ละ T_i ทำให้เห็นว่าโหนดลูกของโหนด r จะมีเพียงโหนดเดียวเท่านั้นและจะต้องแตกต่างกันในทุกๆ T_i ทำให้พิสูจน์ได้ว่า สำหรับเครือข่ายแบบไฮเปอร์คิว k มิติใดๆ จะมีต้นไม้ทอดข้ามที่แตกต่าง T_i $(0 \leq i \leq k\text{-}1)$ ที่มีโหนด r เป็นโหนดรากทั้งหมด k ต้น ตามจำนวนของโหนดลูกของโหนด r บนเครือข่ายแบบไฮเปอร์คิว k มิติ □

ตัวอย่างที่ 4.1 แสดงตัวอย่างของต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ที่มีโหนดรากเป็นโหนด 8 แสดงดังรูปด้านล่าง และการคำนวณหาโหนดลูกของโหนด 8 ใน T_0 ถึง T_3 ของ Q_4 แสดงดังตารางที่ 4.1



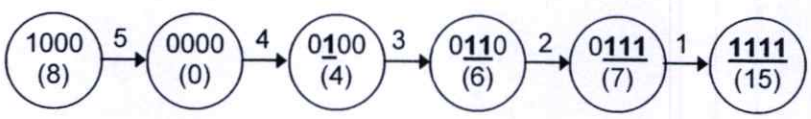
ตารางที่ 4.1 แสดงตัวอย่างการคำนวณหาโหนดลูกของโหนด 8(1000) ตั้งแต่ T_0 ถึง T_3 ของ Q_4

T_i	หมายเลขประจำโหนดของโหนด 8 (รูปแบบเลขฐานสอง)	r_i	โหนดลูกของโหนด 8 (รูปแบบเลขฐานสอง)	โหนดลูกของโหนด 8 (การปรับบิตที่ i ของ 8)
T_0	100***0***	$r_0 = 0$	100***1***	9 $(=8+2^0)$
T_1	10***0***0	$r_1 = 0$	10***1***0	10 $(=8+2^1)$
T_2	1***0***00	$r_2 = 0$	1***1***00	12 $(=8+2^2)$
T_3	***1***000	$r_3 = 1$	***0***000	0 $(=8-2^3)$

ทฤษฎีบทที่ 2 สำหรับเครือข่ายไฮเปอร์คิว k มิติใดๆ ความสูงของต้นไม้ทอดข้ามที่แตกต่าง T_i ($0 \leq i \leq k-1$) ซึ่งมีโหนดรากเป็นโหนด r เป็นความสูงที่เหมาะสมคือมีความสูงไม่เกิน k+1 เส้นเชื่อม

พิสูจน์ จากทฤษฎีบทที่ 1 จะเห็นได้ว่าในแต่ละ T_i โหนดราก r จะมีโหนดลูกซึ่งเป็นโหนดที่อยู่ติดกันคือโหนด $r \pm 2^i$ และเนื่องจากความยาวของเส้นทางที่แตกต่างจากโหนด $x \notin \{r, r \pm 2^i\}$ ไปยังโหนด $r \pm 2^i$ เท่ากับระยะทางแฮมมิ่งระหว่าง x กับ $r \pm 2^i$ ดังนั้นเส้นทางที่ยาวที่สุดจากโหนด $x = d_{k-1}\ldots d_{i+1}x_i d_{i-1}\ldots d_0 \notin \{r, r \pm 2^i\}$ (โดยที่ $x_i = r_i$ และ d_j แทนค่าของบิตในตำแหน่งที่ j เมื่อ $x_j \neq r_j$) ไปยังโหนด $r \pm 2^i$ จึงเท่ากับ k เส้นเชื่อม และเนื่องจากระยะทางแฮมมิ่งระหว่างโหนด r และโหนด $r \pm 2^i$ เท่ากับ 1 เส้นเชื่อม ดังนั้นเส้นทางจากโหนด r จนถึงโหนด $x = d_{k-1}\ldots d_{i+1}x_i d_{i-1}\ldots d_0$ ซึ่งเป็นโหนดที่อยู่ไกลจากโหนด r ที่สุดในเส้นทางจึงเท่ากับ k+1 เส้นเชื่อม จากเหตุผลข้างต้นจึงพิสูจน์ได้ว่า ความสูงของต้นไม้ทอดข้ามที่แตกต่าง T_i ($0 \leq i \leq k-1$) ที่มีโหนดรากเป็นโหนด r ที่สร้างจากขั้นตอนวิธีเป็นความสูงที่เหมาะสมคือมีความสูงไม่เกิน k+1 เส้นเชื่อม □

ตัวอย่างที่ 4.2 พิจารณาเส้นทางที่ยาวที่สุดใน T_3 (ดังรูปที่ 4.1) ที่มีโหนดรากเป็นโหนด 8(1000) บน Q_4 ($k = 4, i = 3, r = 8$) แสดงดังรูปที่ 4.1



รูปที่ 4.1 แสดงเส้นทางที่ยาวที่สุดใน T_3 ที่มีโหนดรากเป็นโหนด 8(1000) บน Q_4

เมื่อพิจารณาเส้นทางบน T_3 จะพบว่าโหนดลูกของโหนด 8 ใน T_3 คือโหนด 0(0000) และโหนดที่ไกลที่สุดในเส้นทางคือโหนด 15 $\notin \{8, 0(=8-2^3)\}$ เมื่อพิจารณาหมายเลขประจำโหนดของโหนด 15 ในรูปแบบของเลขฐานสองจะได้ว่า $x = x_3x_2x_1x_0 = 1111$ โดยที่ $x_3 = r_3 = 1$ จากเลขฐานสองซึ่งเป็นหมายเลขประจำโหนดของทั้งโหนด 15(1111) และโหนด 0(0000) จะพบว่าระยะทางแฮมมิ่งระหว่างทั้ง 2 โหนดนี้เท่ากับ 4 หรืออีกนัยหนึ่งก็คือเส้นทางระหว่างทั้ง 2 โหนดคือ 4 เส้นเชื่อม (= k) และเนื่องจากระยะทางระหว่างโหนด 8 และโหนด 0 เท่ากับ 1 เส้นเชื่อม ทำให้ระยะทางจากโหนด 15 ไปยังโหนด 8 เท่ากับ 5 (= k+1) เส้นเชื่อม

ทฤษฎีบทที่ 3 ในต้นไม้ทอดข้ามที่แตกต่าง T_i $(0 \le i \le k-1)$ ที่มีโหนดรากเป็นโหนด r เส้นทางจากโหนด $x(\neq r)$ ซึ่งเป็นสมาชิกของเครือข่ายแบบไฮเปอร์คิว k มิติ ไปยังโหนด r ในทุกๆ T_i เป็นเส้นทางที่แตกต่างกัน

พิสูจน์ จากขั้นตอนวิธีการสร้างต้นไม้ทอดข้ามที่แตกต่างบนเครือข่ายแบบไฮเปอร์คิว k มิติ ซึ่งประยุกต์ใช้จัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไปเพื่อช่วยหาเส้นทางที่ไม่ซ้ำกันจากโหนด $x(\neq r) \in Q_k$ ไปยังโหนดราก r เมื่อ $x = x_{k-1}x_{k-2}\ldots x_i\ldots x_1x_0$ และ $i \in \{0, 1, \ldots, k-1\}$ โดย $H_{x,r} = \{i_0, i_1, \ldots, i_{t-2}, i_{t-1}\}$ และ $i_0 < i_1 < \ldots < i_{t-2} < i_{t-1}$ โดยสัญลักษณ์ที่ใช้แทนเส้นทางของ Q_k คือ $x \xrightarrow{+2^i} y$ ในกรณีที่ $x + 2^i = y$ หรือ $x \xrightarrow{-2^i} y$ ในกรณีที่ $x - 2^i = y$ ซึ่งมีความหมายว่า x และ y เป็นโหนดที่อยู่ติดกันในเส้นทางนั้นๆ สมมติให้ $i = i_p$ สำหรับ p ใดๆ โดยที่ $p = 0, 1, \ldots, t-1$ ซึ่งทำให้สามารถพิจารณาเส้นทางที่แตกต่างใน T_i ตามกฎทิศทางของบิต (Bit direction rule) ได้ 2 กรณีดังนี้

กรณีที่ 1 : $i \in H_{x,r}$ และ $x_i \neq r_i$ (หรือกรณีที่ x เป็นโหนดภายใน (Internal node) ใน T_i) เนื่องจาก $H_{x,r} = \{i_0, i_1, \ldots, i_p, i_{p+1}, \ldots, i_{t-1}\}$ ดังนั้น x จึงเป็นโหนดประชิดของ $x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}}$ (หรืออีกนัยหนึ่งคือ x เป็นโหนดประชิดของ $x-2^{i_{p+1}}$ ในกรณีที่ $x_{i_{p+1}} = 1$ หรือ x เป็นโหนดประชิดของ $x+2^{i_{p+1}}$ ในกรณีที่ $x_{i_{p+1}} = 0$) ใน T_i กำหนดให้ $y = x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}}$ และเนื่องจาก $H_{y,r} = \{i'_0, i'_1, \ldots, i'_p, i'_{p+1}, \ldots, i'_{t-2}\}$ ดังนั้น y จึงเป็นโหนดประชิดของ $y+(-)^{y_{i'_{p+1}}} 2^{i'_{p+1}}$ ใน T_i และกำหนดให้ $z = y+(-)^{y_{i'_{p+1}}} 2^{i'_{p+1}}$ โดยที่ $H_{z,r} = \{i''_0, i''_1, \ldots, i''_p, i''_{p+1}, \ldots, i''_{t-3}\}$ ดังนั้น z จึงเป็นโหนดประชิดของ $z+(-)^{z_{i''_{p+1}}} 2^{i''_{p+1}}$ ใน T_i ซึ่งทำให้สามารถหาเส้นทางที่แตกต่างที่เชื่อมระหว่าง x และ r ใน T_i ได้ดังนี้

$$x \xrightarrow{(-)^{x_{i_{p+1}}} 2^{i_{p+1}}} (x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}}) \xrightarrow{(-)^{y_{i'_{p+1}}} 2^{i'_{p+1}}} (x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}}+(-)^{y_{i'_{p+1}}} 2^{i'_{p+1}}) \xrightarrow{(-)^{z_{i''_{p+1}}} 2^{i''_{p+1}}} \ldots$$

$$\xrightarrow{(-)^{d_{i^+_{p+1}}} 2^{i^+_{p+1}}} (x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}}+(-)^{y_{i'_{p+1}}} 2^{i'_{p+1}}+\ldots+(-)^{d_{i^+_{p+1}}} 2^{i^+_{p+1}}) \xrightarrow{(-)^{x_i} 2^i} r$$

หมายเหตุ สำหรับโหนด x ในแต่ละ T_i ที่ $x_i \neq r_i$ จะมี $H_{x,r}$ เดียวกัน แต่พิจารณาที่ตำแหน่ง i และ succ(i) ต่างกันจึงมี parent(T, x) ต่างกัน

กรณีที่ 2 : $i \notin H_{x,r}$ และ $x_i = r_i$ (หรือกรณีที่ x เป็นโหนดใบไม้ (Leaf node) ใน T_i) จากกฎทิศทางของบิตที่ว่า x เป็นโหนดประชิดของ $x+(-)^{x_i} 2^i$ (หรืออีกนัยหนึ่งคือ x เป็นโหนดประชิดของ $x-2^i$ ในกรณีที่ $x_i = 1$ หรือ x เป็นโหนดประชิดของ $x+2^i$ ในกรณีที่ $x_i = 0$) ใน T_i กำหนดให้ $y = x+(-)^{x_i} 2^i$ ซึ่งทำให้ $i \in H_{y,r}$ จากกรณีที่ 1 ที่แสดงว่ามีเส้นทางที่แตกต่างที่เชื่อมระหว่าง y และ r ใน

T_i ดังนั้นในกรณีที่ $x_i = 1$ เส้นทางจาก x ไป y คือ $x \xrightarrow{-2^i} y$ หรือในกรณีที่ $x_i = 0$ เส้นทางจาก x ไป y คือ $x \xrightarrow{+2^i} y$ จึงเป็นส่วนหนึ่งของเส้นทางที่แตกต่างซึ่งทำให้ x เชื่อมต่อกับ r ใน T_i

<u>หมายเหตุ</u> สำหรับโหนด x ในแต่ละ T_i ที่ $x_i = r_i$ จะพิจารณาปรับบิตที่ตำแหน่ง i ต่างกัน (i = 0, 1, 2, ..., k-1) จึงมี parent(T, x) ต่างกัน

จากการพิจารณาเส้นทางที่แตกต่างใน T_i ทั้ง 2 กรณีข้างต้น จะเห็นได้ว่าทุกๆ โหนดมีค่า x หรือมีหมายเลขประจำโหนดที่แตกต่างกัน และเมื่อพิจารณา ณ ค่า i ใดๆ เส้นทางจากทุกๆ โหนด $x(\neq r)$ ไปยังโหนด r ใน T_i ย่อมแตกต่างกัน และจากค่า i ที่พิจารณาในแต่ละ T_i ไม่เหมือนกัน ($i \in \{0, 1, 2, \ldots, k-1\}$) เส้นทางจากโหนด $x(\neq r)$ ใดๆ ไปยังโหนด r จึงเป็นเส้นทางที่ไม่ซ้ำกัน จึงสรุปได้ว่าเส้นทาง $T_0[x,r]$, $T_1[x,r]$, ... และ $T_{k-1}[x,r]$ เป็นเส้นทางที่ไม่ซ้ำกัน □

<u>ตัวอย่างที่ 4.3</u> กำหนดให้ r = 8(1000) ในที่นี้จะสามารถพิจารณาเส้นทางที่แตกต่างจากทุกๆ โหนด $x(\neq 8)$ ไปยังโหนด 8 ในต้นไม้ทอดข้ามที่แตกต่าง T_0, T_1, T_2 และ T_3 ของ Q_4 ได้ดังนี้

เส้นทางจาก x = 0000(0) ไปยัง r = 1000(8)

$T_0[0, 8]$: 0:0000 $\xrightarrow{+2^0}$ 1:0001 $\xrightarrow{+2^3}$ 9:1001 $\xrightarrow{-2^0}$ 8:1000

$T_1[0, 8]$: 0:0000 $\xrightarrow{+2^1}$ 2:0010 $\xrightarrow{+2^3}$ 10:1010 $\xrightarrow{-2^1}$ 8:1000

$T_2[0, 8]$: 0:0000 $\xrightarrow{+2^2}$ 4:0100 $\xrightarrow{+2^3}$ 12:1100 $\xrightarrow{-2^2}$ 8:1000

$T_3[0, 8]$: 0:0000 $\xrightarrow{+2^3}$ 8:1000

เส้นทางจาก x = 0001(1) ไปยัง r = 1000(8)

$T_0[1, 8]$: 1:0001 $\xrightarrow{+2^3}$ 9:1001 $\xrightarrow{-2^0}$ 8:1000

$T_1[1, 8]$: 1:0001 $\xrightarrow{+2^1}$ 3:0011 $\xrightarrow{+2^3}$ 11:1011 $\xrightarrow{-2^0}$ 10:1010 $\xrightarrow{-2^1}$ 8:1000

$T_2[1, 8]$: 1:0001 $\xrightarrow{+2^2}$ 5:0101 $\xrightarrow{+2^3}$ 13:1101 $\xrightarrow{-2^0}$ 12:1100 $\xrightarrow{-2^2}$ 8:1000

$T_3[1, 8]$: 1:0001 $\xrightarrow{-2^0}$ 0:0000 $\xrightarrow{+2^3}$ 8:1000

เส้นทางจาก x = 0010(2) ไปยัง r = 1000(8)

$T_0[2, 8]$: 2:0010 $\xrightarrow{+2^0}$ 3:0011 $\xrightarrow{-2^1}$ 1:0001 $\xrightarrow{+2^3}$ 9:1001 $\xrightarrow{-2^0}$ 8:1000

$T_1[2, 8]$: 2:0010 $\xrightarrow{+2^3}$ 10:1010 $\xrightarrow{-2^1}$ 8:1000

$T_2[2, 8]$: 2:0010 $\xrightarrow{+2^2}$ 6:0110 $\xrightarrow{+2^3}$ 14:1110 $\xrightarrow{-2^1}$ 12:1100 $\xrightarrow{-2^2}$ 8:1000

$T_3[2, 8]$: 2:0010 $\xrightarrow{-2^1}$ 0:0000 $\xrightarrow{+2^3}$ 8:1000

เส้นทางจาก x = 0011(3) ไปยัง r = 1000(8)

$T_0[3, 8]$: $3{:}00\mathbf{11} \xrightarrow{-2^1} 1{:}\mathbf{0}001 \xrightarrow{+2^3} 9{:}100\mathbf{1} \xrightarrow{-2^0} 8{:}1000$

$T_1[3, 8]$: $3{:}\mathbf{0}011 \xrightarrow{+2^3} 11{:}101\mathbf{1} \xrightarrow{-2^0} 10{:}10\mathbf{1}0 \xrightarrow{-2^1} 8{:}1000$

$T_2[3, 8]$: $3{:}0011 \xrightarrow{+2^2} 7{:}\mathbf{0}111 \xrightarrow{+2^3} 15{:}111\mathbf{0} \xrightarrow{-2^0} 14{:}11\mathbf{1}0 \xrightarrow{-2^1} 12{:}1\mathbf{1}00 \xrightarrow{-2^2} 8{:}1000$

$T_3[3, 8]$: $3{:}001\mathbf{1} \xrightarrow{-2^0} 2{:}00\mathbf{1}0 \xrightarrow{-2^1} 0{:}\mathbf{0}000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 0100(4) ไปยัง r = 1000(8)

$T_0[4, 8]$: $4{:}010\mathbf{0} \xrightarrow{+2^0} 5{:}0\mathbf{1}01 \xrightarrow{-2^2} 1{:}\mathbf{0}001 \xrightarrow{+2^3} 9{:}100\mathbf{1} \xrightarrow{-2^0} 8{:}1000$

$T_1[4, 8]$: $4{:}01\mathbf{0}0 \xrightarrow{+2^1} 6{:}0\mathbf{1}10 \xrightarrow{-2^2} 2{:}\mathbf{0}010 \xrightarrow{+2^3} 10{:}10\mathbf{1}0 \xrightarrow{-2^1} 8{:}1000$

$T_2[4, 8]$: $4{:}\mathbf{0}100 \xrightarrow{+2^3} 12{:}1\mathbf{1}00 \xrightarrow{-2^2} 8{:}1000$

$T_3[4, 8]$: $4{:}0\mathbf{1}00 \xrightarrow{-2^2} 0{:}\mathbf{0}000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 0101(5) ไปยัง r = 1000(8)

$T_0[5, 8]$: $5{:}0\mathbf{1}01 \xrightarrow{-2^2} 1{:}\mathbf{0}001 \xrightarrow{+2^3} 9{:}100\mathbf{1} \xrightarrow{-2^0} 8{:}1000$

$T_1[5, 8]$: $5{:}01\mathbf{0}1 \xrightarrow{+2^1} 7{:}0\mathbf{1}11 \xrightarrow{-2^2} 3{:}\mathbf{0}011 \xrightarrow{+2^3} 11{:}101\mathbf{1} \xrightarrow{-2^0} 10{:}10\mathbf{1}0 \xrightarrow{-2^1} 8{:}1000$

$T_2[5, 8]$: $5{:}\mathbf{0}101 \xrightarrow{+2^3} 13{:}110\mathbf{1} \xrightarrow{-2^0} 12{:}1\mathbf{1}00 \xrightarrow{-2^2} 8{:}1000$

$T_3[5, 8]$: $5{:}010\mathbf{1} \xrightarrow{-2^0} 4{:}0\mathbf{1}00 \xrightarrow{-2^2} 0{:}\mathbf{0}000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 0110(6) ไปยัง r = 1000(8)

$T_0[6, 8]$: $6{:}011\mathbf{0} \xrightarrow{+2^0} 7{:}01\mathbf{1}1 \xrightarrow{-2^1} 5{:}0\mathbf{1}01 \xrightarrow{-2^2} 1{:}\mathbf{0}001 \xrightarrow{+2^3} 9{:}100\mathbf{1} \xrightarrow{-2^0} 8{:}1000$

$T_1[6, 8]$: $6{:}0\mathbf{1}10 \xrightarrow{-2^2} 2{:}\mathbf{0}010 \xrightarrow{+2^3} 10{:}10\mathbf{1}0 \xrightarrow{-2^1} 8{:}1000$

$T_2[6, 8]$: $6{:}\mathbf{0}110 \xrightarrow{+2^3} 14{:}11\mathbf{1}0 \xrightarrow{-2^1} 12{:}1\mathbf{1}00 \xrightarrow{-2^2} 8{:}1000$

$T_3[6, 8]$: $6{:}01\mathbf{1}0 \xrightarrow{-2^1} 4{:}0\mathbf{1}00 \xrightarrow{-2^2} 0{:}\mathbf{0}000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 0111(7) ไปยัง r = 1000(8)

$T_0[7, 8]$: $7{:}01\mathbf{1}1 \xrightarrow{-2^1} 5{:}0\mathbf{1}01 \xrightarrow{-2^2} 1{:}\mathbf{0}001 \xrightarrow{+2^3} 9{:}100\mathbf{1} \xrightarrow{-2^0} 8{:}1000$

$T_1[7, 8]$: $7{:}0\mathbf{1}11 \xrightarrow{-2^2} 3{:}\mathbf{0}011 \xrightarrow{+2^3} 11{:}101\mathbf{1} \xrightarrow{-2^0} 10{:}10\mathbf{1}0 \xrightarrow{-2^1} 8{:}1000$

$T_2[7, 8]$: $7{:}\mathbf{0}111 \xrightarrow{+2^3} 15{:}111\mathbf{1} \xrightarrow{-2^0} 14{:}11\mathbf{1}0 \xrightarrow{-2^1} 12{:}1\mathbf{1}00 \xrightarrow{-2^2} 8{:}1000$

$T_3[7, 8]$: $7{:}011\mathbf{1} \xrightarrow{-2^0} 6{:}01\mathbf{1}0 \xrightarrow{-2^1} 4{:}0\mathbf{1}00 \xrightarrow{-2^2} 0{:}\mathbf{0}000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 1001(9) ไปยัง r = 1000(8)

$T_0[9, 8]$: $9{:}1001 \xrightarrow{-2^0} 8{:}1000$

$T_1[9, 8]$: $9{:}1001 \xrightarrow{+2^1} 11{:}1011 \xrightarrow{-2^0} 10{:}1010 \xrightarrow{-2^1} 8{:}1000$

$T_2[9, 8]$: $9{:}1001 \xrightarrow{+2^2} 13{:}1101 \xrightarrow{-2^0} 12{:}1100 \xrightarrow{-2^2} 8{:}1000$

$T_3[9, 8]$: $9{:}1001 \xrightarrow{-2^3} 1{:}0001 \xrightarrow{-2^0} 0{:}0000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 1010(10) ไปยัง r = 1000(8)

$T_0[10, 8]$: $10{:}1010 \xrightarrow{+2^0} 11{:}1011 \xrightarrow{-2^1} 9{:}1001 \xrightarrow{-2^0} 8{:}1000$

$T_1[10, 8]$: $10{:}1010 \xrightarrow{-2^1} 8{:}1000$

$T_2[10, 8]$: $10{:}1010 \xrightarrow{+2^2} 14{:}1110 \xrightarrow{-2^1} 12{:}1100 \xrightarrow{-2^2} 8{:}1000$

$T_3[10, 8]$: $10{:}1010 \xrightarrow{-2^3} 2{:}0010 \xrightarrow{-2^1} 0{:}0000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 1011(11) ไปยัง r = 1000(8)

$T_0[11, 8]$: $11{:}1011 \xrightarrow{-2^1} 9{:}1001 \xrightarrow{-2^0} 8{:}1000$

$T_1[11, 8]$: $11{:}1011 \xrightarrow{-2^0} 10{:}1010 \xrightarrow{-2^1} 8{:}1000$

$T_2[11, 8]$: $11{:}1011 \xrightarrow{+2^2} 15{:}1111 \xrightarrow{-2^0} 14{:}1110 \xrightarrow{-2^1} 12{:}1100 \xrightarrow{-2^2} 8{:}1000$

$T_3[11, 8]$: $11{:}1011 \xrightarrow{-2^3} 3{:}0011 \xrightarrow{-2^0} 2{:}0010 \xrightarrow{-2^1} 0{:}0000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 1100(12) ไปยัง r = 1000(8)

$T_0[12, 8]$: $12{:}1100 \xrightarrow{+2^0} 13{:}1101 \xrightarrow{-2^2} 9{:}1001 \xrightarrow{-2^0} 8{:}1000$

$T_1[12, 8]$: $12{:}1100 \xrightarrow{+2^1} 14{:}1110 \xrightarrow{-2^2} 10{:}1010 \xrightarrow{-2^1} 8{:}1000$

$T_2[12, 8]$: $12{:}1100 \xrightarrow{-2^2} 8{:}1000$

$T_3[12, 8]$: $12{:}1100 \xrightarrow{-2^3} 4{:}0100 \xrightarrow{-2^2} 0{:}0000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 1101(13) ไปยัง r = 1000(8)

$T_0[13, 8]$: $13{:}1101 \xrightarrow{-2^2} 9{:}1001 \xrightarrow{-2^0} 8{:}1000$

$T_1[13, 8]$: $13{:}1101 \xrightarrow{+2^1} 15{:}1111 \xrightarrow{-2^2} 11{:}1011 \xrightarrow{-2^0} 10{:}1010 \xrightarrow{-2^1} 8{:}1000$

$T_2[13, 8]$: $13{:}1101 \xrightarrow{-2^0} 12{:}1100 \xrightarrow{-2^2} 8{:}1000$

$T_3[13, 8]$: $13{:}1101 \xrightarrow{-2^3} 5{:}0101 \xrightarrow{-2^0} 4{:}0100 \xrightarrow{-2^2} 0{:}0000 \xrightarrow{+2^3} 8{:}1000$

เส้นทางจาก x = 1110(14) ไปยัง r = 1000(8)

$T_0[14, 8]$: 14:1110 $\xrightarrow{+2^0}$ 15:1111 $\xrightarrow{-2^1}$ 13:1101 $\xrightarrow{-2^2}$ 9:1001 $\xrightarrow{-2^0}$ 8:1000

$T_1[14, 8]$: 14:1110 $\xrightarrow{-2^2}$ 10:1010 $\xrightarrow{-2^1}$ 8:1000

$T_2[14, 8]$: 14:1110 $\xrightarrow{-2^1}$ 12:1100 $\xrightarrow{-2^2}$ 8:1000

$T_3[14, 8]$: 14:1110 $\xrightarrow{-2^3}$ 6:0110 $\xrightarrow{-2^1}$ 4:0100 $\xrightarrow{-2^2}$ 0:0000 $\xrightarrow{+2^3}$ 8:1000

เส้นทางจาก x = 1111(15) ไปยัง r = 1000(8)

$T_0[15, 8]$: 15:1111 $\xrightarrow{-2^1}$ 13:1101 $\xrightarrow{-2^2}$ 9:1001 $\xrightarrow{-2^0}$ 8:1000

$T_1[15, 8]$: 15:1111 $\xrightarrow{-2^2}$ 11:1011 $\xrightarrow{-2^0}$ 10:1010 $\xrightarrow{-2^1}$ 8:1000

$T_2[15, 8]$: 15:1111 $\xrightarrow{-2^0}$ 14:1110 $\xrightarrow{-2^1}$ 12:1100 $\xrightarrow{-2^2}$ 8:1000

$T_3[15, 8]$: 15: 1111 $\xrightarrow{-2^3}$ 7:0111 $\xrightarrow{-2^0}$ 6:0110 $\xrightarrow{-2^1}$ 4:0100 $\xrightarrow{-2^2}$ 0:0000 $\xrightarrow{+2^3}$ 8:1000

จากการพิสูจน์ทฤษฎีบทที่ 3 ทั้ง 2 กรณี และตัวอย่างที่ 4.3 จะเห็นว่าเส้นทางจากโหนด $x(\neq r)$ ไปยังโหนด r ในทุกๆ T_i ไม่ซ้ำกัน เนื่องจากค่า i ที่พิจารณาในที่นี้คือ 0 ถึง k-1 แสดงถึงการพิจารณาหาเส้นทางที่ไม่ซ้ำในทุกๆ T_i ตั้งแต่ T_0 ถึง T_{k-1}

บทที่ 5
บทสรุป

เนื้อหาในบทนี้แบ่งออกเป็น 2 ส่วนด้วยกันโดยในส่วนแรกจะเป็นการสรุปเนื้อหาของงานวิจัย และในส่วนที่ 2 จะเป็นการเสนอแนวทางในการพัฒนางานวิจัยเกี่ยวกับการสร้างต้นไม้ทอดข้ามที่แตกต่าง

5.1 บทสรุป

งานวิจัยนี้ได้นำเสนอวิธีการแบบทั่วไปในการสร้างต้นไม้ทอดข้ามที่แตกต่างด้วยขั้นตอนวิธีแบบขนานบนเครือข่ายแบบไฮเปอร์คิว ซึ่งขั้นตอนวิธีการดังกล่าวปรับปรุงมาจากขั้นตอนวิธีแบบขนานในการสร้างต้นไม้ทอดข้ามบนเครือข่ายแบบไฮเปอร์คิวซึ่งเฉพาะเจาะจงที่โหนดรากเป็นโหนด 0 โดยใช้หลักการของจัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไป (Generalized Hamming Distance Latin Square : GHDLS) จากหลักการนี้ทำให้สามารถสร้างต้นไม้ทอดข้ามที่แตกต่างซึ่งมีโหนดรากเป็นโหนด r (r = 0, 1, 2, …, หรือ 2^k-1) บนเครือข่ายแบบไฮเปอร์คิว k มิติได้ด้วยวิธีการแบบขนาน

สำหรับเครือข่ายแบบไฮเปอร์คิว k มิติ (Q_k) ใดๆ จะสามารถสร้างต้นไม้ทอดข้ามที่แตกต่าง T_i ($0 \leq i \leq k-1$) ที่มีโหนดรากเป็นโหนด r ได้ทั้งหมด k ต้น หากพิจารณาหมายเลขประจำโหนดของ r และ $x(\neq r) \in Q_k$ ในรูปแบบของเลขฐานสอง หมายเลขประจำโหนดในรูปแบบของเลขฐานสองของ $r = r_{k-1}r_{k-2}\ldots r_0$ และ $x = x_{k-1}x_{k-2}\ldots x_0$ จะสามารถสร้างต้นไม้ทอดข้ามที่แตกต่าง k ต้นได้โดยการให้ทุกๆ โหนด x ทำการคำนวณหาหมายเลขประจำโหนดของโหนดพ่อแม่ (Parent determination) ในแต่ละ T_i ตั้งแต่ T_0 จนถึง T_{k-1} ซึ่งแต่ละโหนด x จะทำการคำนวณหาโหนดพ่อแม่โดยเปรียบเทียบหมายเลขประจำโหนด $x_{k-1}x_{k-2}\ldots x_0$ กับ $r_{k-1}r_{k-2}\ldots r_0$ ตั้งแต่บิตที่ 0 จนถึงบิต k-1 และทำการปรับบิต x_i ให้เป็นบิตตรงกันข้ามในกรณีที่ $x_i = r_i$ และในกรณีที่ $x_i \neq r_i$ ให้ทำการปรับบิตต่างตำแหน่งถัดจาก i (succ(i)) ให้เป็นบิตตรงกันข้าม ซึ่งตำแหน่งของบิตต่างตำแหน่งถัดจาก i สามารถพิจารณาได้จากจัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไปของ x แล r เมื่อทุกๆ โหนด x ทราบโหนดพ่อแม่ ก็จะทำให้ทราบเส้นทางที่แตกต่างในแต่ละ T_i ได้

จากการประยุกต์ใช้จัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไปในขั้นตอนวิธี ซึ่งทำให้สร้างต้นไม้ทอดข้ามที่แตกต่างได้ด้วยวิธีการแบบขนาน จึงทำให้ขั้นตอนวิธีดังกล่าวสามารถสร้างต้นไม้ทอดข้ามที่แตกต่างได้ด้วยความซับซ้อนด้านเวลาเท่ากับ O(k) โดยที่ k คือจำนวนมิติของเครือข่ายแบบไฮเปอร์คิว นอกจากนี้ต้นไม้ทอดข้ามที่แตกต่างที่เป็นผลลัพธ์ของขั้นตอนวิธียังมี

ความสูงของต้นไม้ที่เหมาะสมคือ มีความสูงไม่เกิน k+1 เส้นเชื่อม ซึ่งมีผลให้ความยาวเฉลี่ยของเส้นทางในต้นไม้ทอดข้ามที่แตกต่างแต่ละต้นมีค่าน้อยที่สุด

5.2 แนวทางการพัฒนางานวิจัย

นำหลักการของจัตุรัสลาตินของระยะทางแฮมมิ่งแบบทั่วไปไปประยุกต์ใช้ในการสร้างต้นไม้ทอดข้ามที่แตกต่างบนเครือข่ายอื่นๆ ได้แก่

- การสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนดใดๆ บนเครือข่ายทอรัสแบบหลายมิติ (Multidimensional tori network)
- การสร้างต้นไม้ทอดข้ามที่แตกต่างที่มีโหนดรากเป็นโหนดใดๆ บนเครือข่ายแบบคอร์ดอลริง (Chordal rings network)

เป็นต้น

เอกสารอ้างอิง

[1] Bell, J. 2005. "An introduction to SDR's and Latin squares." **Morehead Electronic Journal of Applicable Mathematics**. Vol 4.

[2] Grama, A. Gupta, A. Karypis, G. and Kumar, V. 2003. **Introduction to Parallel Computing.** 2nd ed. Addison Wesley.

[3] Hasunuma, T. 2002. "Completely Independent Spanning Trees in Maximal Planar Graphs." 235–245. **Lecture Notes In Computer Science**. Vol. 2573.

[4] Huck, A. 1994. "Independent Trees in Graphs." 29-45. **Graphs and Combinatorics**. Vol. 10.

[5] Huck, A. 1999. "Independent Trees in Planar Graphs." 29-77. **Graphs and Combinatorics.** Vol. 15.

[6] Iwasaki, Y. Kajiwara, Y. Obokata, K. and Igarashi, Y. 1999. "Independent Spanning Trees of Chordal Rings." 155-160. **Information processing Letters**. Vol.39.

[7] Obokata, K. Iwasaki, Y. Bao, F. and Igarashi, Y. 1996. "Independent Spanning Trees of Product Graphs." 338-351. **Lecture Notes in Computer Science**. Vol.1197.

[8] Petagon, R. and Werapun, J. 2008. **"**Parallel Construction of Independent Spanning Trees on Dynamic Interconnection Networks**."** 187-192. **International Joint conference on computer science and Software Engineering (JCSSE 2008)**. Vol. 1: International Sessions.

[9] Ramanathan, P. and Shin, K.G. 1988. "Reliable Broadcast in Hypercube Multicomputers." 1654-1657. **IEEE Transactions on Computers**. Vol. 37.

[10] Saad, Y. and Schultz, M.H. 1988. "Topological Properties of Hypercube." 867-872. **IEEE Transactions on Computers**. Vol. 37.

[11] Tang, S.M. Wang, Y.L. and Leu, Y.H. 2004. "Optimal Independent Spanning Trees on Hypercubes." 143-155. **Journal of Information Science and Engineering**. Vol. 20.

[12] Tang, S.M. Yang, J.S. Chang, J.M. and Wang, Y.L. 2007. "Parallel Construction of Independent Spanning Trees on Multidimensional Tori." 85–93. **Proceedings of The 24th workshop on combinatorial mathematics and computation theory (CMCT'07)**. Nantou.

[13] Yang, J.S. Chang, J.M. Tang, S.M. and Wang, Y.L. 2007. "Parallel Construction of Optimal Independent Spanning Trees on Hypercubes." 73-79. **ScienceDirect Parallel Computing**. Vol. 33. No. 1.

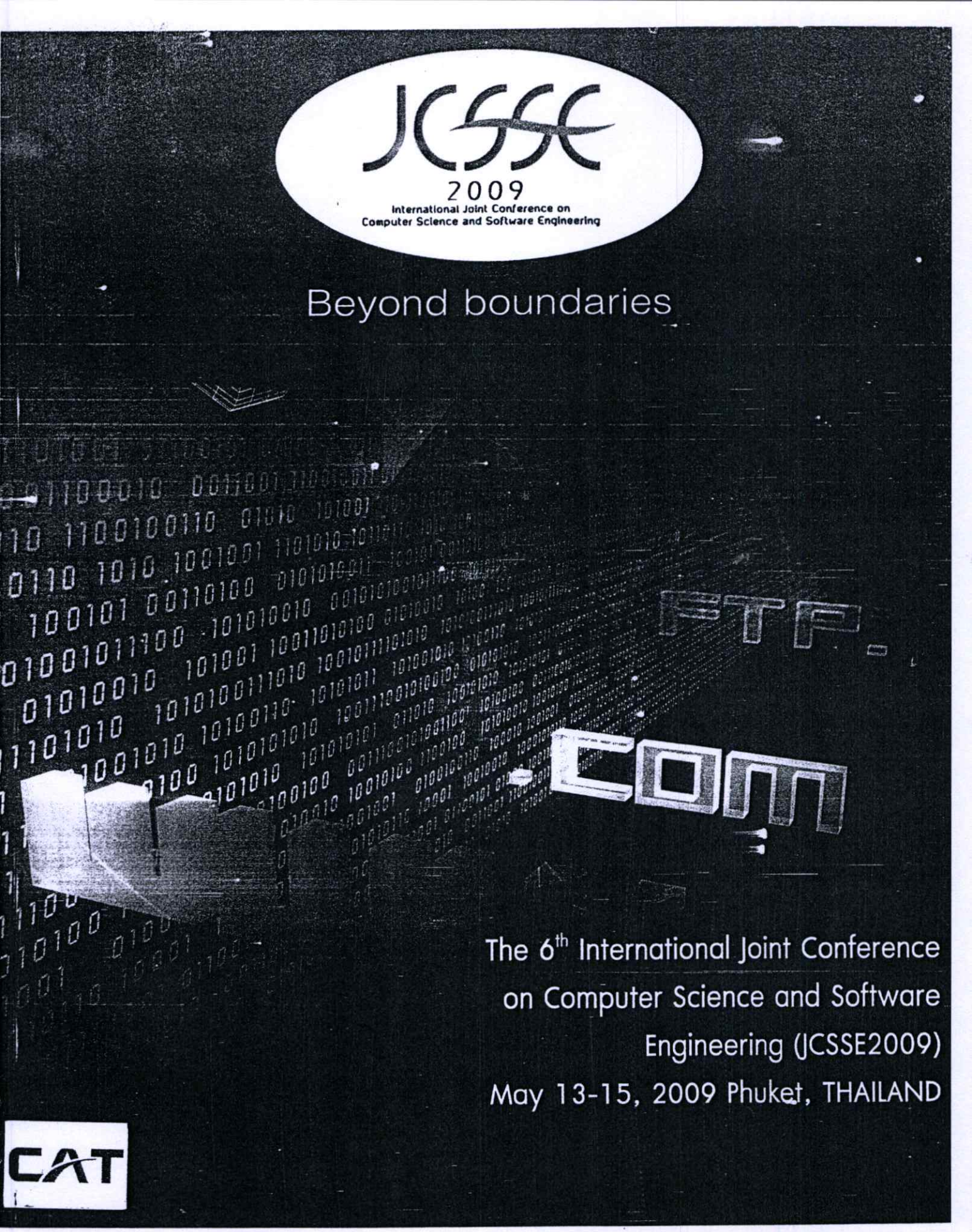
ภาคผนวก

ผลงานที่ได้รับการตีพิมพ์

JCSSE 2009

International Joint Conference on Computer Science and Software Engineering

Beyond boundaries



The 6th International Joint Conference on Computer Science and Software Engineering (JCSSE2009)

May 13-15, 2009 Phuket, THAILAND

CAT

COMGRAPH

Microsoft

COMPUTER SCIENCE
THAMMASAT UNIVERSITY

NECTEC
a member of NSTDA

IEEE
THAILAND SECTION

DWell Group
Company Limited

BANGKOK
ACMSIGGRAPH

Generalized Parallel Construction of Optimal Independent Spanning Trees on Hypercubes

Sumitra Khajonngoen and Jeeraporn Werapun
Department of Computer Science, Faculty of Science,
King Mongkut's Institute of Technology Ladkrabang (KMITL)
Bangkok, 10520, Thailand
Email: s0067507@kmitl.ac.th, ksjeerap@kmitl.ac.th

Abstract

An advantage of independent spanning trees (ISTs) for data broadcasting on any parallel network is the increase of network reliability. Recently work proposed a parallel algorithm for constructing the k optimal ISTs rooted at vertex 0 on k-dimensional (k-D) hypercubes in O(k) time. In this paper, we proposed a generalized parallel algorithm that constructs the k optimal ISTs on k-D hypercubes, based on the Generalized Hamming Distance Latin Square (GHDLS) scheme. Our new parallel algorithm can construct k ISTs rooted at an arbitrary vertex (0, 1, 2, ..., or 2^k-1) in O(k) time and minimize the height of ISTs.

Key Words: Independent spanning trees (*IST*s), Hypercubes, Generalized Hamming Distance Latin square

1. Introduction

Graphs play an important role in parallel computing system by means of describing network topology for resolving many network problems. One of the most interesting problems is data broadcasting on parallel systems. A great number of researches focused on the increase of network fault-tolerance (or reliability). An approach to overcome the problem is achieved by sending multiple copies of the message through disjoint paths [6].

Considering a graph $G = (V,E)$, where $V(G)$ and $E(G)$ denote the vertex set and the edge set of G, respectively. Two paths P and Q connecting between two vertices are said to be internally disjoint, if $E(P) \cap E(Q) = \varnothing$ and $V(P) \cap V(Q)$ is the set of the two end vertices. A spanning tree T of G is a tree composed of all vertices of G, $V(T) = V(G)$, and some edges ($|V(G)|$-1) of G. Two spanning trees T and T' are said to be independent spanning trees (*IST*s) [8] of G, if both T and T' are rooted at the same vertex (r), and for every vertices $v \neq r$, two paths from v to r in both trees are pairwise internally disjoint.

A hypercube, a type of parallel network, is academic and commercial available and is being used for a variety of parallel applications because of their attractive features, including simple recursive-structure, logarithmic diameter, suitability for developing many parallel algorithms, and more importantly "fault-tolerance".

A k-dimensional (k-D) hypercube system (Q_k) is a loosely coupled multiprocessor that can be represented as a graph, consisting of $N = 2^k$ vertices. Each vertex in the hypercube can be uniquely represented by using a k-length binary string ($b_{k-1} \ldots b_1 b_0$). The binary strings of the adjacent vertices differ exactly one place, and hence each vertex has k-directions to connect to adjacent vertices. Generally, the k-D hypercube can be constructed recursively from lower dimensional hypercube i.e. $Q_k = Q_{k-1} \times Q_1$, where is Q_1 is a complete graph with two vertices only and '$\times$' is the Cartesian product operation [7].

For a distributed computing network (i.e., Q_k), broadcasting on a graph can be realized based on the *IST*. Meanwhile, the design of multiple *IST*s has been applied to fault-tolerance protocols. For example, a fault-tolerance broadcasting protocol in Q_k can be designed by means of k *IST*s. With the root r, a source node, fault-tolerance is achieved by sending k copies of messages through the k *IST*s. If the source node is the non-faulty node, this scheme can tolerate up to k-1 faulty nodes.

The problem of constructing multiple *IST*s has been studied to design the optimal *IST*s for a given graph has received more attention, such as planar graphs [1,2,3], product graphs [5], chordal rings [4], multidimensional tori [9], and hypercubes [8,10], etc. Most of researches on this problem tended to pursue two research goals. The first is to invent the efficient constructing *IST* algorithm in terms of time complexity and the second is to reduce the *IST*s heights. In 1988, Ramanathan and Shin [6] introduced an algorithm for the reliable broadcasting in any hypercube, called the recursive doubling algorithm. That algorithm generates disjoint paths in a top-down manner (root to leaves) based on the bit-direction rule. The basic idea is the source node sends the message to all

of its neighborhoods. Then, the neighborhoods in turn forward the received message to their adjacent nodes through the generated disjoint paths. In 2004, Tang et al. [8] proposed an algorithm for constructing the height reduced *IST*s for hypercubes. Their algorithm was modified from [5] to improve the optimal result in sense of the average path length. In recently work, Yang et al. [10] proposed an algorithm to generate the optimal *IST*s, on k-dimensional (k-D) hypercubes, based on "Hamming distance Latin square" scheme. By this scheme, the algorithm can be parallelized in $O(k)$ time and the resulting *IST*s are optimal in respect of the average path length. The existing algorithms consider vertex 0 as the root of the *IST*s since the hypercube (Q_k) is vertex-symmetric [5,8,10]. However, every vertex (or node) in Q_k will be assigned node's ID at once (for initial setting) and cannot be relocated to other different IDs during runtime. Therefore, under this scheme we can assign any vertex to be the root (= 0) for broadcasting but after allocation time (or mapping nodes' IDs) we cannot perform the broadcasting from the root (r) that is not zero ($r \neq 0$).

In this paper, we propose a parallel algorithm to construct the k *IST*s rooted at any vertex (v) on the k-D hypercubes (Q_k). In our new algorithm, any vertex in Q_k can be the root for broadcasting at any time. The idea of our generalized parallel algorithm with any root is extended from the parallel algorithm with a specific root ($r = 0$) [10], which is optimal in terms of the average path length, and hence our results are also optimal.

The remainder of this paper is organized as follows: Section 2 introduces the related work, the algorithm for constructing the k optimal *IST*s for Q_k root at 0. Section 3 proposes our generalized parallel algorithm for constructing the k optimal *IST*s rooted at an arbitrary vertex on Q_k and the proof of correctness is provided in Section 4. Finally, the last section contains the conclusion of our work.

2. Related work

In this section, we would like to introduce the recently algorithm that constructs the k optimal *IST*s for Q_k. Before presenting the algorithm, some notations and useful properties are defined as follows: For a tree T rooted at a vertex $r = 0$, let x ($\neq r$) is a vertex in T, the parent of x, denoted by *parent*(T, x), is a vertex adjacent to x in the path $T[r, x]$.

Let vertex x ($\neq r$) $\in Q_k$ with a binary string $x = x_{k-1} \ldots x_2 x_1 x_0$. A Hamming distance Latin square (*HDLS*) [10] with respect to x is a Latin square whose elements are chosen from the set $H_x = \{i: 0 \leq i \leq k\text{-}1 \text{ and } x_i = 1\}$, the set of locations of 1-bits in x. Suppose that $x(\neq 0)$ has Hamming distance t to the root (r=0) and let $H_x = \{i_0, i_1, \ldots, i_j, \ldots, i_{t-1}\}$ be the 1st row of the *HDLS*, such that $i_0 < i_1 \ldots < i_j < \ldots < i_{t-1}$. With respect to x in Q_k, the matrix shown below is called the increasingly rotational *HDLS*. In addition, if i_j is an element in $I_k(x)$ then the *successor* of i_j is i_{j+1} and the *predecessor* of i_j is i_{j-1} are denoted by succ(i_j) and pred(i_j), respectively.

$$I_k(x) = \begin{bmatrix} i_0 & i_1 & \cdots & i_{t-2} & i_{t-1} \\ i_1 & i_2 & \cdots & i_{t-1} & i_0 \\ \vdots & & \cdots & & \\ i_{t-2} & i_{t-1} & \cdots & i_{t-4} & i_{t-3} \\ i_{t-1} & i_0 & \cdots & i_{t-3} & i_{t-2} \end{bmatrix}$$

Example 1: We consider the vertex x = 1111(15). $H_x = \{i_0, i_1, \ldots, i_{t-1}\} = \{0, 1, 2, 3\}$ and the corresponding *HDLS* matrix defined above is given by

$$I_4(15) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$$

From the above matrix, if we consider i_j = 2 then succ(2) is 3 and pred(2) is 1.

The main idea of this algorithm is to construct a set $\tau = \{T_i : 0 \leq i \leq k\text{-}1\}$ of k *IST*s rooted at vertex r= 0 for Q_k in a bottom-up manner (leaves to root). In parallel, every vertex (except the root) determines its parent in each tree T_i by comparing the binary string of every vertex $x(\neq 0)$ to vertex 0 bit by bit (from bit 0 to bit k-1) and then adjusting bit weight on the binary string of each vertex to generate its parent in each T_i.

We rewrite the algorithm proposed by Yang et al. [10] as follows:

Algorithm GEN-PARENTS
begin
for all $x(\neq 0) \in Q_k$ with binary string $x = x_{k-1}x_{k-2} \ldots x_0$ **pardo**
  for i = 0 to k-1 **do**
    if (x_i=0) *parent*(T_i,x) = $x+2^i$; // adjust bit weight
    if (x_i=1) *parent*(T_i,x) = $x-2^{\text{succ}(i)}$; // apply HDLS
  end for i
end for all
end GEN-PARENTS

Example 2: Constructing T_0 for Q_4 is computed as follows. Let $x(\neq 0)$ be any vertex in Q_4 with binary string $x = x_3x_2x_1x_0$. In case of $x_0 = 0$, leaf vertices $x \in \{2, 4, 6, 8, 10, 12, 14\}$ and then *parent*(T_0,x) = $x+2^0$ = {3, 5, 7, 9, 11, 13, 15}. The remaining cases (internal vertices) are defined as follows: if x = 1 then succ(0) = 0 and *parent*(T_0,1) = $1-2^{\text{succ}(0)} = 1-2^0 = 0$; if $x \in \{3, 7, 11, 15\}$ then succ(0) = 1 and *parent*(T_0,x) = $x-2^1$ = {1, 5, 9, 13}; if $x \in \{5, 13\}$ then succ(0) = 2 and *parent*(T_0,x) = $x-2^2$ = {1, 9}. Finally, x = 9 then succ(0) = 3 and *parent*(T_0,9) = $9-2^3$ = 1. Note: the construction of T_1, T_2, and T_3 for Q_4 can be generated similarly to that of T_0 (Figure 1).

According to the *HDLS* concept, the algorithm can be parallelized and the construction of k *IST*s is simple. By this way, Yang et al. show that the path from $x(\neq 0)$ to 0 in each T_i are pairwise internally disjoint. Moreover, the height of T_i is at most $k+1$ since the length of the unique path from $x \notin \{0, 2^i\}$ to the vertex 2^i equals to the Hamming distance between x and 2^i. The longest path from $x = 11..1x_i1..1 \notin \{0, 2^i\}$ to the vertex 2^i, where $x_i = 0$, requires k edges, and hence $k+1$ edges along the path to the root ($r = 0$).

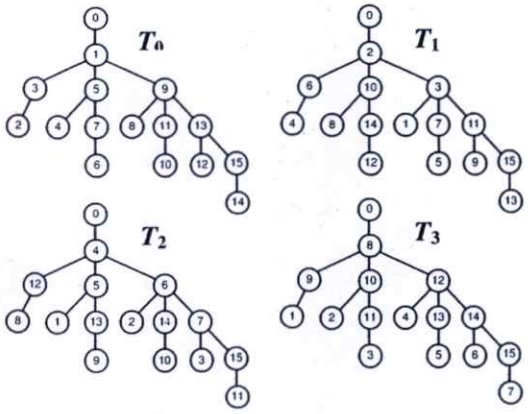
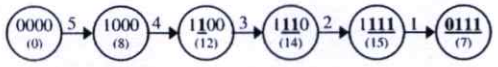


Figure 1. A set of *IST*s rooted at vertex 0 on Q_4.

Example 3: Considering the longest path in T_3 (Figure 1) that rooted at vertex 0 on Q_4 ($k = 4$, $i = 3$, $r = 0$), as follows:



In T_3, the child of $r = 0$ is 8 (1000) and the farthest vertex is $x = 7 \notin \{0, 8\}$ then $x = x_3x_2x_1x_0 = 0111$ and x_3=0. The Hamming distance between $x = 7$ and 8 (child of r) is 4 then the path length between them is 4(=k) edges, and required 5(=k+1) edges along the path to the root.

Although any hypercube (Q_k) is vertex-symmetric, (any vertex can assign ID = 0 and setting the root of *IST*s = 0 is possible), every vertex in Q_k can be assigned node's ID only once and cannot be relocated to other IDs during runtime. Therefore, one limitation of the existing algorithms [5,8,10] is that only vertex 0 can be the root of *IST*s after allocating IDs.

3. Proposed generalized algorithm

In this section, we propose the generalized algorithm for constructing k optimal *IST*s rooted at any vertex (0, 1, 2,…,or 2^k-1) for Q_k, modified from the study in [10], the specific root at vertex 0. For convenience, we first introduce some notations and useful concepts. Similar to the previous section, for a tree T rooted at a vertex r, let x ($\neq r$) is a vertex in T, the parent of x in T, denoted by *parent*(T,x). Suppose $\tau = \{T_i: 0 \le i \le k-1\}$ is a set of k *IST*s rooted at r for Q_k. By the connectivity of Q_k and the definition of τ, there is only one unique child of r, which is $r+2^i$ in case of $r_i = 0$ or $r-2^i$ in case of $r_i = 1$, in each T_i.

Furthermore, we introduce the Generalized Hamming distance Latin square (*GHDLS*) that is extended from the *HDLS* [10]. Let vertex r, as the source node, is the root of *IST*s and vertex $x(\neq r) \in Q_k$ with binary string $r = r_{k-1}\ldots r_2r_1r_0$ and $x = x_{k-1}\ldots x_2x_1x_0$. The *GHDLS* is a Latin square whose entries are chosen from the set $H_{x,r} = \{i: 0 \le i \le k-1$ and $x_i \neq r_i\}$, the set of locations of different bits in x and r. Suppose that x has Hamming distance t to r, let $H_{x,r} = \{i_0, i_1, \ldots, i_j, \ldots, i_{t-1}\}$ be the 1st row of the *GHDLS*, such that $i_0 < i_1 \ldots < i_j < \ldots < i_{t-1}$. With respect to x and r in Q_k, the increasing *GHDLS*, denoted by $I_k(x,r)$, is the following matrix. From the matrix, if i_j is an entry in $I_k(x,r)$ then i_{j-1} is called the predecessor of i_j, denoted by pred(i_j), and the successor of i_j is i_{j+1}, denoted by succ(i_j).

$$I_k(x,r) = \begin{bmatrix} i_0 & i_1 & \cdots & i_{t-2} & i_{t-1} \\ i_1 & i_2 & \cdots & i_{t-1} & i_0 \\ \vdots & & \cdots & & \\ i_{t-2} & i_{t-1} & \cdots & i_{t-4} & i_{t-3} \\ i_{t-1} & i_0 & \cdots & i_{t-3} & i_{t-2} \end{bmatrix}$$

Example 4: Considering Q_4, if r = 1000(8) be the root and the vertex x = 1111(15), the set $H_{15,8} = \{0, 1, 2\}$, then the generated *GHDLS* matrix of x is shown in the following figure.

$$I_4(15,8) = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

From the matrix, in case of $i_j = 2$, hence 0 is the successor of 2, or succ(2), and the predecessor of 2, or pred(2), is 1.

Let's see another case, for the same r = 1000(8), if vertex x = 0111(7) then $H_{7,8} = \{0, 1, 2, 3\}$ and the generated matrix is illustrated as follow. From matrix $I_4(7,8)$, if we consider $i_j = 2$ then succ(2) is 3 and pred(2) is 1.

$$I_4(7,8) = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$$

According to the *GHDLS* concept, the construction of k *IST*s with any root (r) can be parallelized on Q_k in O(k) time. Our proposed parallel algorithm is detailed as follows:

Algorithm PARALLEL GEN-PARENTs-in-ISTs
begin
  for all vertex $x(\neq r) \in Q_k$ **pardo**
    with $r = r_{k-1}r_{k-2}\ldots r_0$, $x = x_{k-1}x_{k-2}\ldots x_0$
    for $i = 0$ to k-1 **do**
      if ($r_i = x_i$) // adjust bit weight
        if ($x_i = 0$) $parent(T_i,x) = x + 2^i$
        if ($x_i = 1$) $parent(T_i,x) = x - 2^i$
      if ($r_i \neq x_i$) // apply GHDLS
        if ($x_{succ(i)} = 0$) $parent(T_i,x) = x + 2^{succ(i)}$
        if ($x_{succ(i)} = 1$) $parent(T_i,x) = x - 2^{succ(i)}$
    end for i
  end for all
end PARALLEL GEN-PARENTs-in-ISTs

Example 5: We illustrate constructing T_0 and T_3 rooted at r, with binary string $r = r_3r_2r_1r_0$, for Q_4, suppose the source node is vertex 8 then r = 1000(8). Let $x(\neq 8)$, with binary string $x = x_3x_2x_1x_0$, be any vertex in Q_4.

Four *ISTs* (T_0, T_1, T_2, T_3) rooted at vertex 8 of Q_4 are depicted in Figure 2. In addition, Table 1 and 2 show the complete information about the construction of T_0 and T_3 rooted at vertex 8 for Q_4, respectively.

For constructing T_0 (Figure 2), if x_0=r_0=0, leaf vertices $x \in$ {0, 2, 4, 6, 10, 12, 14} then in this case, we have $parent(T_0, x) = x+2^0$ = {1, 3, 5, 7, 11, 13, 15}. The remaining cases (internal vertices), $x_0 \neq r_0$, are the following: if x = 9 then succ(0)=0 and $parent(T_0, 9) = 9 - 2^0 = 8$; if $x \in$ {3, 7, 11, 15} then succ(0) = 1 and $x_{succ(0)} = 1$, so we have $parent(T_0,x) = x - 2^1$ = {1, 5, 9, 13}; if $x \in$ {5, 13} then succ(0) = 2, $x_{succ(0)} = 1$ and $parent(T_0,x) = x-2^2$ = {1,9}; if x=1 then succ(0) = 3, $x_{succ(0)} = 0$ and $parent(T_0,1) = 1+2^3 = 9$.

For constructing T_3 (Figure 2), if $x_3 = r_3 = 1$, leaf vertices $x \in$ {9, 10, 11, 12, 13, 14, 15} then in this case, we have $parent(T_3,x) = x - 2^3$ = {1, 2, 3, 4, 5, 6, 7}. The other cases (internal vertices), $x_3 \neq r_3$, are the following: if $x \in$ {1, 3, 5, 7} then succ(3) = 0, $x_{succ(3)} = 1$ and $parent(T_3,x) = x - 2^0$ = {0, 2, 4, 6}; if $x \in$ {2, 6} then succ(3) = 1, $x_{succ(3)} = 1$ and $parent(T_3,x) = x - 2^1$ = {0, 4}; if x = 4 then succ(3) = 2 and $x_{succ(3)} = 1$, so we have $parent(T_3,4) = 4 - 2^2 = 0$. Finally, x = 0 then succ(3) = 3, $x_{succ(3)} = 0$ and $parent(T_3,0) = 0+2^3 = 8$.

4. Proof of correctness

In this section, we provide the proof of correctness in "the optimal height of k *ISTs* (T_0, T_1, ..., T_{k-1}) and "the unique paths in each T_i". According to the proposed algorithm, since two end vertices of any edge in T_i are with Hamming distance 1, obviously the child that adjacent to the root (r) in each T_i are obtained by adjusting bit weight of the root from bit 0 to bit k-1 ($r+2^i$ in case of $r_i = 0$ or $r-2^i$ in case of $r_i = 1$) and hence there exist k *ISTs* in Q_k. From Example 5, the child of vertex 8 in each T_i, can be obtained by adjusting bit weight of 8(1000) from bit i ($r+2^i$ in case of $r_i = 0$ or $r-2^i$ in case of $r_i = 1$) , i = 0 to 3, are shown in Table 3.

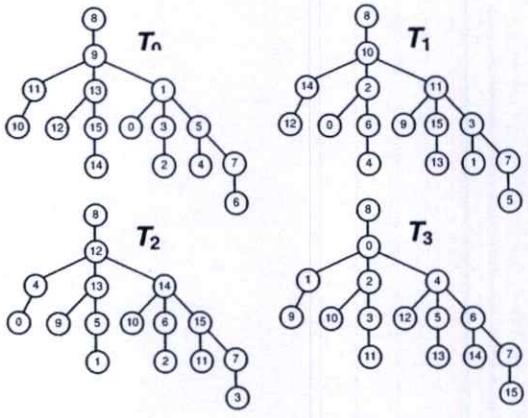


Figure 2. A set of *ISTs* rooted at vertex 8 on Q_4.

Table 1. The parent of a vertex $x(\neq 8) \in Q_4$ in T_0.

x	Binary string	succ(0)	$2^{succ(0)}$	$x_{succ(0)}$	$parent(T_0,x)$
0	000*0*	-	-	-	1
1	*0*001	3	8	0	9
2	001*0*	-	-	-	3
3	00*1*1	1	2	1	1
4	010*0*	-	-	-	5
5	0*1*01	2	4	1	1
6	011*0*	-	-	-	7
7	01*1*1	1	2	1	5
9	100*1*	0	1	1	8
10	101*0*	-	-	-	11
11	10*1*1	1	2	1	9
12	110*0*	-	-	-	13
13	1*1*01	2	4	1	9
14	111*0*	-	-	-	15
15	11*1*1	1	2	1	13

Table 2. The parent of a vertex $x(\neq 8) \in Q_4$ in T_3.

x	Binary string	succ(3)	$2^{succ(3)}$	$x_{succ(3)}$	$parent(T_3,x)$
0	*0*000	3	8	0	8
1	000*1*	0	1	1	0
2	00*1*0	1	2	1	0
3	001*1*	0	1	1	2
4	0*1*00	2	4	1	0
5	010*1*	0	1	1	4
6	01*1*0	1	2	1	4
7	011*1*	0	1	1	6
9	*1*001	-	-	-	1
10	*1*010	-	-	-	2
11	*1*011	-	-	-	3
12	*1*100	-	-	-	4
13	*1*101	-	-	-	5
14	*1*110	-	-	-	6
15	*1*111	-	-	-	7

Table 3. The child of the root (vertex 8) in each T_i.

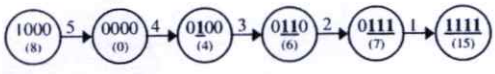
T_i	Binary string of vertex 8	r_i	Child of vertex 8 (binary string)	Child of vertex 8 (adjust bit weight)
T_0	1000	$r_0 = 0$	1001	9 (=8+2^0)
T_1	1000	$r_1 = 0$	1010	10 (=8+2^1)
T_2	1000	$r_2 = 0$	1100	12 (=8+2^2)
T_3	1000	$r_3 = 1$	0000	0 (=8-2^3)

4.1 The optimal height of k *ISTs* in Q_k

First, we will show that "the height of T_i is at most k+1" since the length of the unique path from $x \notin \{r, r\pm2^i\}$ to the vertex $r\pm2^i$ (child of r) equals to the Hamming distance between x and $r\pm2^i$.

The longest path from $x = d_{k-1}..d_{i+1}x_id_{i-1}..d_0 \notin \{r, r\pm2^i\}$ to the vertex $r\pm2^i$ requires k edges, and hence k+1 edges along the path to the root (r), where $x_i = r_i$, and d_j be a different bit between x_j & r_j of x and r. Therefore, our resulting *ISTs* are optimal in the sense of average path length, similar to [10].

Example 6: Considering the longest path in T_3 (Figure 2) that rooted at vertex r = 8 (1000) on Q_4 (k = 4, i = 3, r = 8), as follows:



In T_3, The child of 8 is 0 (0000) and the farthest vertex is x = 15 $\notin$ {8, 0(=8-2^3)} then $x = x_3x_2x_1x_0$ = 1111 and $x_3 = r_3$ = 1. The Hamming distance between x = 15 and 0 (child of r) is 4 then the path length between them is 4(=k) edges, and required 5(=k+1) edges along the path to the root.

4.2 The unique paths of k *ISTs* in Q_k

Next, we will show that the path from $x(\neq r)$ to r in each T_i are pairwise internally disjoint. Clearly, we can find the unique path from every vertex $x(\neq r)$ to r, with binary string $r = r_{k-1}r_{k-2}...r_0$ and $x = x_{k-1}x_{k-2}...x_0$, based on the *GHDLS*. From the definition of $H_{x,r}$, suppose that $H_{x,r}$ = $\{i_0, i_1, \ldots, i_p, \ldots, i_{t-1}\}$ with $i_0 < i_1 <...< i_p <...< i_{t-1}$ and notations $x \xrightarrow{+2^i} y$ or $x \xrightarrow{-2^i} y$, means that x and y are adjacent in the path.

Assume $i = i_p$, for some p = 0, 1, ... , t-1. For T_i, we can consider the unique paths, according to the bit direction rule in two cases, as follows:

Case 1: $i \in H_{x,r}$ and $x_i \neq r_i$ (or internal vertices (x) in T_i). Since $H_{x,r} = \{i_0, i_1, \ldots, i_p, i_{p+1}, \ldots, i_{t-1}\}$, x is adjacent to $x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}}$ (or x-$2^{i_{p+1}}$ if x_i=1; otherwise x+$2^{i_{p+1}}$) . Let $y = x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}}$ due to $H_{y,r} = \{i'_0, i'_1, \ldots, i'_p, i'_{p+1}, \ldots, i'_{t-2}\}$, y is adjacent to $y+(-)^{y_{i_{p+1}}} 2^{i'_{p+1}}$, where i'_{p+1} = succ(i'_p) = i_{p+2}. Let $z = y+(-)^{y_{i_{p+1}}} 2^{i'_{p+1}}$ then $H_{z,r} = \{i''_0, i''_1,\ldots, i''_p, i''_{p+1},\ldots, i''_{t-3}\}$, i''_{p+1}= succ(i''_p)=i_{p+3}. Similarly, z is adjacent to $z+(-)^{z_{i_{p+1}}} 2^{i''_{p+1}}$, and so on. In this case, the unique path that connecting x to r in T_i is

$$x \xrightarrow{(-)^{x_{i_{p+1}}} 2^{i_{p+1}}} (x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}})$$

$$\xrightarrow{(-)^{y_{i_{p+1}}} 2^{i'_{p+1}}} (x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}} +(-)^{y_{i_{p+1}}} 2^{i'_{p+1}})$$

$$\xrightarrow{(-)^{z_{i_{p+1}}} 2^{i''_{p+1}}} \ldots \xrightarrow{(-)^{d_{i^+_{p+1}}} 2^{i^+_{p+1}}} (x+(-)^{x_{i_{p+1}}} 2^{i_{p+1}} +(-)^{y_{i_{p+1}}} 2^{i_{p+1}}$$

$$+ \ldots +(-)^{d_{i^+_{p+1}}} 2^{i^+_{p+1}}) \xrightarrow{(-)^{x_i} 2^i} r.$$

Case 2: $i \notin H_{x,r}$ and $x_i = r_i$ (or leaf vertices (x) in T_i). By the bit direction rule, x is adjacent to $x+(-)^{x_i} 2^i$ (or x-2^i if x_i = 1; otherwise x+2^i). Let $y = x+(-)^{x_i} 2^i$ then $i \in H_{y,r}$. From Case 1, we have already shown that there is the unique path connecting y and r in T_i. Therefore $x \xrightarrow{-2^i} y$ (if x_i = 1) or $x \xrightarrow{+2^i} y$ (if x_i = 0) is along with the unique path connecting x to r.

Example 7: For Q_4, there are four *ISTs* that rooted at vertex 8 (Figure 2). Let's consider the unique paths from vertex 6 (x = 6:0110) to root 8 (r = 8:1000) in every tree.

For T_0[6,8], the 1st unique path from 6→8 is derived as follows: In T_0, $x_0 = r_0 = 0$ ($\notin H_{6,8}$). x = 6 is adjacent to y = x+2^0 = 7 (Case 2).

$$\begin{array}{l} 6{:}0110 \xrightarrow{-2^0} 7{:}0111 \\ 8{:}1000 \\ (ddd\text{-}) \end{array}$$

where - represent $x_i = r_i$ and d represent $x_i \neq r_i$.

Next, from y = 7 to r = 8 (1000), y = 7 (0111) is adjacent to $z = y - 2^1$ = 5 (Case 1: $y_0 \neq r_0$, y_0 = 1 and i_{p+1} = 1) and so on for other vertices along that unique path from 5→8.

$$\begin{array}{lll} 7{:}0111 & \xrightarrow{-2^1} & 5{:}0101 \xrightarrow{-2^2} 1{:}0001 \\ 8{:}1000 & & 8{:}1000 \\ (dddd), i_{p+1}=1 & & (dd\text{-}d), i'_{p+1}=2 \end{array}$$

$$\begin{array}{lll} 1{:}0001 & \xrightarrow{+2^3} & 9{:}1001 \xrightarrow{-2^0} 8{:}1000 \\ 8{:}1000 & & 8{:}1000 \\ (d\text{--}d), i''_{p+1}=3 & & (\text{---}d), i'''_{p+1}=0 \end{array}$$

Therefore, the complete unique path from 6 → 8 in T_0 is

T_0[6,8]: $6 \xrightarrow{-2^0} 7 \xrightarrow{-2^1} 5 \xrightarrow{-2^2} 1 \xrightarrow{+2^3} 9 \xrightarrow{-2^0} 8.$

Similarly, other unique paths from $6 \rightarrow 8$ in T_1, T_2, T_3 are

$$T_1[6,8] : 6 \xrightarrow{-2^2} 2 \xrightarrow{+2^3} 10 \xrightarrow{-2^1} 8$$
(Case 1: i=1, (6→8))

$$T_2[6,8] : 6 \xrightarrow{+2^3} 14 \xrightarrow{-2^1} 12 \xrightarrow{-2^2} 8$$
(Case 1: i=2, (6→8))

$$T_3[6,8] : 6 \xrightarrow{-2^1} 4 \xrightarrow{-2^2} 0 \xrightarrow{-2^3} 8$$
(Case 1: i=3, (6→8))

5. Conclusion

In this paper, we propose a generalized parallel algorithm for constructing k optimal *IST*s with any root (0, 1, 2, ..., or 2^k-1) for Q_k, that consists of $N = 2^k$ vertices, based on the Generalized Hamming distance Latin square (*GHDLS*) scheme. Our optimal algorithm can be performed in $O(k)$ time, where k is a number of hypercube dimensions. Furthermore, the farthest distance from a vertex (x) to the root (r), refer to the height of the *IST*, is at most k+1 and hence our resulting *IST*s are optimal in the sense of average path length.

Acknowledgment

The authors are thankful to King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, THAILAND for the academic and the research financial support.

6. References

[1] T. Hasunuma, "Completely Independent Spanning Trees in Maximal Planar Graphs", Lecture Notes In Computer Science, Vol. 2573, 2002, pp.235-245.

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

[3] A. Huck, "Independent trees in planar graphs", Graphs and Combinatorics, Vol. 15, 1999, pp.29-77.

[4] Y. Iwasaki, Y. Kajiwara, K. Obokata, and Y. Igarashi, "Independent spanning trees of chordal rings", Information processing Letters, Vol.39, 1999, pp.155-160.

[5] K. Obokata, Y. Iwasaki, F. Bao, and Y. Igarashi, "Independent spanning trees of product graphs", Lecture Notes in Computer Science, Vol.1197, 1996, pp.338-351.

[6] P. Ramanathan and K. G. Shin, "Reliable broadcast in hypercube multicomputers", IEEE Transactions on Computers, Vol. 37, 1988, pp.1654-1657.

[7] Y. Saad and M. H. Schultz, "Topological properties of hypercube", IEEE Transactions on Computers, Vol. 37, 1988, pp.867-872.

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

[9] S. M. Tang, J. S. Yang, J. M. Chang, and Y. L. Wang, "Parallel construction of independent spanning trees on multidimensional tori", Proceedings of The 24th workshop on combinatorial mathematics and computation theory (CMCT'07), Nantou, Taiwan, 2007, pp.085-093.

[10] J. S. Yang, J. M. Chang, S. M. Tang, and Y. L. Wang, "Parallel construction of optimal independent spanning trees on hypercubes", ScienceDirect Parallel Computing, Vol. 33, No. 1, 2007, pp.73-79

ประวัติผู้เขียน

ชื่อ – สกุล	นางสาวสุมิตรา ขจรเงิน
วัน เดือน ปีเกิด	2 มกราคม 2521
ที่อยู่	65/58 หมู่บ้านพฤกษาวิลล์ 3 ซอยเทพประทาน ถ.สรงประภา แขวงสีกัน เขตดอนเมือง จังหวัดกรุงเทพมหานคร 10210

ประวัติการศึกษา

2542	จบการศึกษาปริญญาวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ (ประสานมิตร)
2550- ปัจจุบัน	กำลังศึกษาวิทยาศาสตรมหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาศาสตร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประวัติการทำงาน

2545 - ปัจจุบัน	ตำแหน่งนักวิชาการคอมพิวเตอร์ หน่วยงานเทคโนโลยีสารสนเทศ ศูนย์การแพทย์สมเด็จพระเทพฯ คณะแพทยศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ