

Lecture 8

เอกสารประกอบการบรรยาย Data Structure เรื่อง Tree

Tree

เนื้อหา

- โครงสร้างข้อมูลแบบทรี
- นิยามของทรี
- นิยามที่เกี่ยวข้องกับทรี
- การแทนที่ทรีในหน่วยความจำหลัก
- การแปลงทรีทั่วไปให้เป็นไบนารีทรี

Tree (Cont.)

เนื้อหา (ต่อ)

- การท่องไปในทรี
- เอ็กซ์เพรสชันทรี
- ไบนารีเซิร์ชทรี

จุดประสงค์การเรียนรู้

1. เพื่อให้ นักศึกษาทราบโครงสร้างข้อมูลแบบทรี
2. เพื่อให้ทราบนิยามของทรี และที่เกี่ยวข้อง
3. เพื่อให้ นักศึกษาทราบวิธีการแทนที่ทรีในหน่วยความจำหลัก
4. เพื่อให้ นักศึกษาทราบการแปลงทรีให้เป็นไบนารีทรี
5. เพื่อให้ นักศึกษาทราบวิธีการท่องไปในทรี

จุดประสงค์การเรียนรู้

6. เพื่อให้นักศึกษาทราบกระบวนการ
เอ็กซ์เพรสชันทรี
7. เพื่อให้ศึกษาทราบวิธีการไบนารีเซิร์ชทรี

Tree (Cont.)

ทรี (Tree) เป็นโครงสร้างข้อมูลที่มีความสัมพันธ์
ระหว่าง โหนดจะมีความสัมพันธ์ลดหลั่นกันเป็นลำดับ
ชั้น (Hierarchical Relationship)
ได้มีการนำรูปแบบทรีไปประยุกต์ใช้ในงาน
ต่าง ๆ อย่างแพร่หลาย ส่วนมากจะใช้สำหรับแสดง
ความสัมพันธ์ระหว่างข้อมูล

เช่น แผนผังองค์ประกอบของหน่วยงานต่าง ๆ
โครงสร้างสารบัญหนังสือ เป็นต้น

Tree (Cont.)

แต่ละโหนดจะมีความสัมพันธ์กับโหนดใน
ระดับที่ต่ำลงมา หนึ่งระดับได้หลาย ๆ โหนด
เรียกโหนดดังกล่าวว่า **โหนดแม่ (Parent or
Mother Node)**

โหนดที่อยู่ต่ำกว่าโหนดแม่อยู่หนึ่งระดับ
เรียกว่า **โหนดลูก (Child or Son Node)**

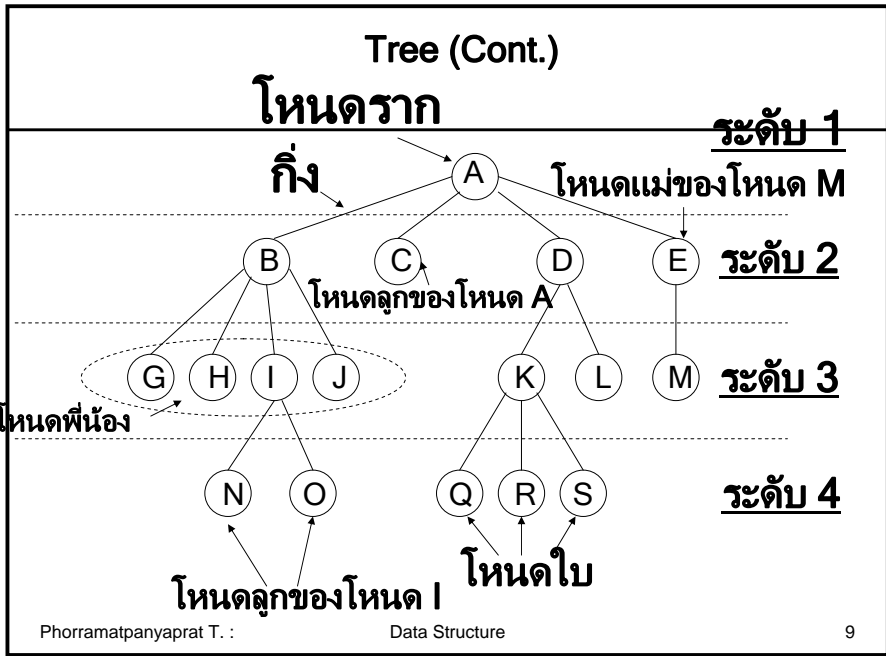
โหนดที่อยู่ในระดับสูงสุดและไม่มีโหนดแม่
เรียกว่า **โหนดราก (Root Node)**

Tree (Cont.)

โหนดที่มีโหนดแม่เป็นโหนดเดียวกัน
เรียกว่า **โหนดพี่น้อง (Siblings)**

โหนดที่ไม่มีโหนดลูก เรียกว่า
โหนดใบ (Leave Node)

เส้นเชื่อมแสดงความสัมพันธ์ระหว่าง
โหนดสองโหนด
เรียกว่า **กิ่ง (Branch)**



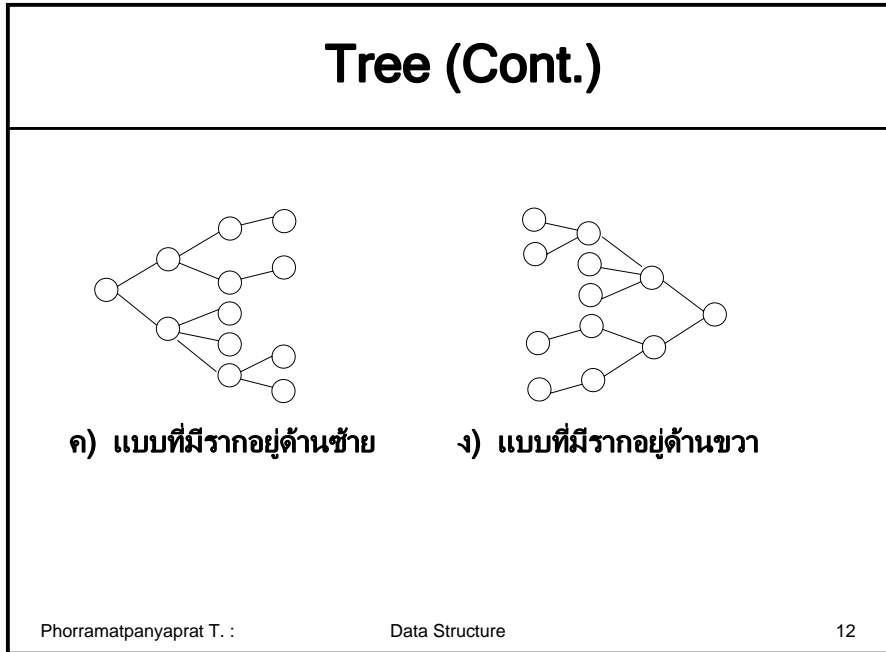
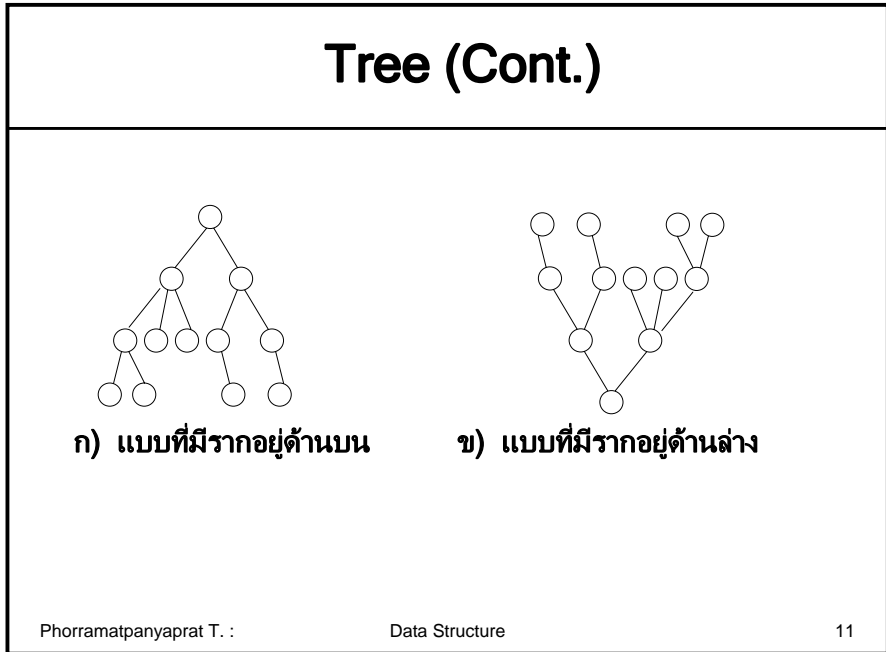
Tree (Cont.)

นิยามของทรี

1. นิยามทรีด้วยนิยามของกราฟ
 ทรี คือ กราฟที่ต่อเนื่องโดยไม่มีวงจรปิด (loop) ในโครงสร้าง โหนดสองโหนดใด ๆ ในทรีต้องมีทางติดต่อกันทางเดียวเท่านั้น และทรีที่มี N โหนด ต้องมีกิ่งทั้งหมด N-1 เส้น

การเขียนรูปแบบทรี อาจเขียนได้ 4 แบบ คือ

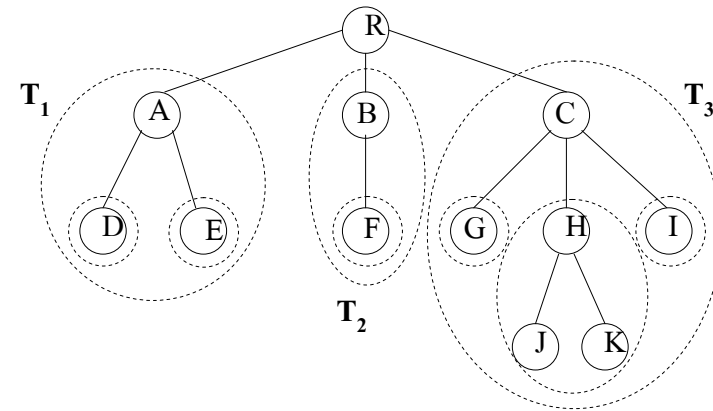
Phorramatpanyaprat T. : Data Structure 10



Tree (Cont.)

2. นิยามทรีด้วยรูปแบบรีเคอร์ซีฟ
ทรีประกอบด้วยสมาชิกที่เรียกว่า
โหนด โดยที่ ถ้าว่าง ไม่มีโหนดใดๆ เรียกว่า
นัลทรี (Null Tree) และถ้ามีโหนด
หนึ่งเป็นโหนดราก ส่วนที่เหลือจะแบ่งเป็น
ทรีย่อย (Sub Tree)
 $T_1, T_2, T_3, \dots, T_k$ โดยที่ $k \geq 0$ และทรีย่อย
ต้องมีคุณสมบัติเป็นทรี

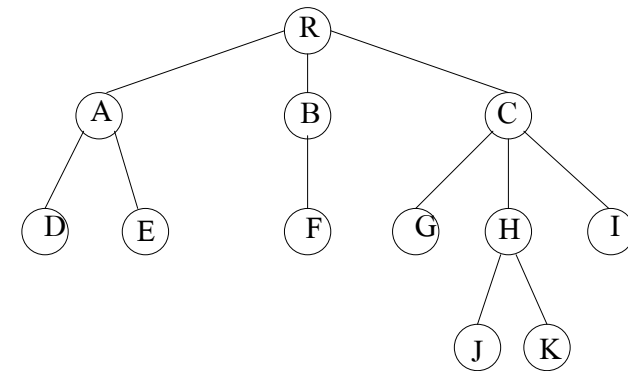
Tree (Cont.)



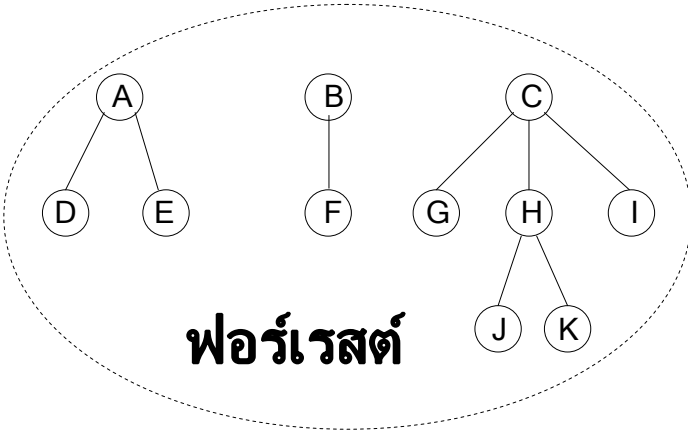
Tree (Cont.)

นิยามที่เกี่ยวข้องกับทรี
1. ฟอว์เรสต์ (Forest)
หมายถึง กลุ่มของทรีที่เกิดจากการ
เอาโหนดรากของทรีออก
หรือ เซตของทรีที่แยกจากกัน
(Disjoint Trees)

Tree (Cont.)

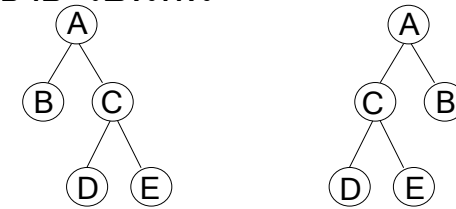


Tree (Cont.)



Tree (Cont.)

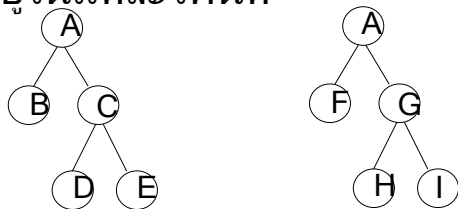
2. ทรีที่มีแบบแผน (Ordered Tree)
หมายถึง ทรีที่โหนดต่างๆ ในทรีนั้นมี
ความสัมพันธ์ที่แน่นอน เช่น ไปทางขวา
ไปทางซ้าย เป็นต้น



แสดงทรีที่ต่างกันเมื่อกำหนดให้เป็นทรีที่มีแบบแผน

Tree (Cont.)

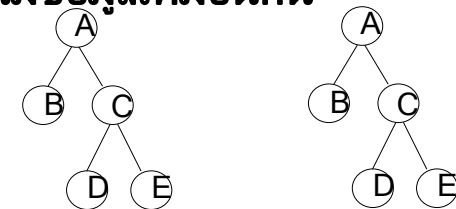
3. ทรีคล้าย (Similar Tree) คือ
ทรีที่มีโครงสร้างเหมือนกัน หรือทรีที่มี
รูปร่างของทรีเหมือนกัน โดยไม่คำนึงถึง
ข้อมูลที่อยู่ในแต่ละโหนด



แสดงทรีที่คล้ายกัน

Tree (Cont.)

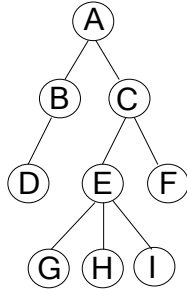
4. ทรีเหมือน (Equivalent Tree) คือ
ทรีที่เหมือนกันโดยสมบูรณ์ โดยต้องเป็นทรีที่
คล้ายกันและแต่ละโหนดในตำแหน่ง
เดียวกันมีข้อมูลเหมือนกัน



แสดงทรีที่เหมือนกัน

Tree (Cont.)

5. กำลัง (Degree) หมายถึง จำนวนทรีย่อยของโหนด นั้น ๆ เช่น



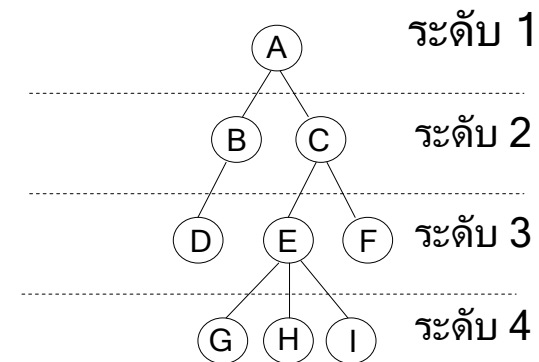
Tree (Cont.)

ในรูปโหนด “B” มีกำลังเป็น 1 เพราะ มีทรีย่อย คือ {“D”}
ส่วนโหนด “C” มีค่ากำลังเป็นสอง เพราะ มีทรีย่อย คือ {“E”, “G”, “H”, “I”} และ {“F”}

Tree (Cont.)

6. ระดับของโหนด (Level of Node) คือ ระยะทางในแนวดิ่งของโหนดนั้น ๆ ที่อยู่ห่างจากโหนดราก เมื่อกำหนดให้ โหนดรากของทรินั้นอยู่ระดับ 1 และกิ่งแต่ละกิ่งมีความเท่ากันหมด คือ ยาวเท่ากับ 1 หน่วย ซึ่งระดับของโหนดจะเท่ากับจำนวนกิ่งที่น้อยที่สุดจากโหนดรากไปยังโหนดใด ๆ บวกด้วย 1 และจำนวนเส้นทางตามแนวดิ่งของโหนดใด ๆ ซึ่งห่างจากโหนดราก เรียกว่า *ความสูง* (Height) หรือ *ความลึก* (Depth)

Tree (Cont.)



แสดงระดับของทรี่

Tree (Cont.)

การแทนที่ทรีในหน่วยความจำหลัก

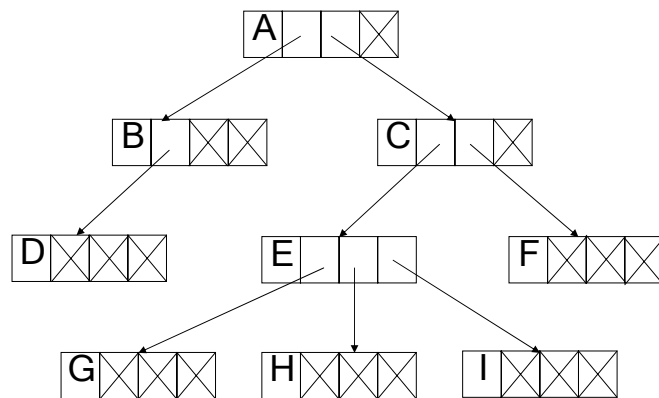
การแทนที่โครงสร้างข้อมูลแบบทรีในหน่วยความจำหลัก จะมีพอยเตอร์เชื่อมโยงจากโหนดแม่ไปยังโหนดลูก แต่ละโหนดต้องมีลิงค์ฟิลด์เพื่อเก็บที่อยู่ของโหนดลูกต่าง ๆ นั่นคือจำนวน ลิงค์ฟิลด์ของแต่ละโหนดขึ้นอยู่กับจำนวนของโหนดลูก

การแทนที่ทรี ซึ่งแต่ละโหนดมีจำนวนลิงค์ฟิลด์ไม่เท่ากัน ทำให้ยากต่อการปฏิบัติการ วิธีการแทนที่ที่ง่ายที่สุดคือ ทำให้แต่ละโหนดมี จำนวนลิงค์ฟิลด์เท่ากัน โดยอาจใช้วิธีการต่อไปนี้

Tree (Cont.)

1. โหนดแต่ละโหนดเก็บพอยเตอร์ชี้ไปยังโหนดลูกทุกโหนด การแทนที่ทรีด้วยวิธีนี้ จะให้จำนวนฟิลด์ในแต่ละโหนดเท่ากันโดยกำหนดให้มีขนาดเท่ากับจำนวนโหนดลูกของโหนดที่มีลูกมากที่สุด โหนดใดไม่มีโหนดลูกก็ให้ค่าพอยเตอร์ในลิงค์ฟิลด์นั้นมีค่าเป็น Null และให้ลิงค์ฟิลด์แรกเก็บค่าพอยเตอร์ชี้ไปยังโหนด ลูกลำดับที่หนึ่ง ลิงค์ฟิลด์ที่สองเก็บค่าพอยเตอร์ชี้ไปยังโหนดลูกลำดับที่สอง และลิงค์ฟิลด์อื่นเก็บค่าพอยเตอร์ของโหนดลูกลำดับ ถัดไปเรื่อย ๆ

Tree (Cont.)



แสดงการแทนที่ทรีซึ่งแต่ละโหนดเก็บพอยเตอร์ชี้ไปยังโหนดลูก

Tree (Cont.)

การแทนทรีด้วยโหนดขนาดเท่ากันค่อนข้างใช้เนื้อที่จำนวนมาก เนื่องจากแต่ละโหนดมีจำนวนโหนดลูกไม่เท่ากันหรือบางโหนดไม่มีโหนดลูกเลย

ถ้าเป็นทรีที่แต่ละโหนดมีจำนวนโหนดลูกที่แตกต่างกันมาก จะเป็นการสิ้นเปลืองเนื้อที่ในหน่วยความจำโดยเปล่าประโยชน์

Tree (Cont.)

2. แทนทรีด้วยไบนารีทรี

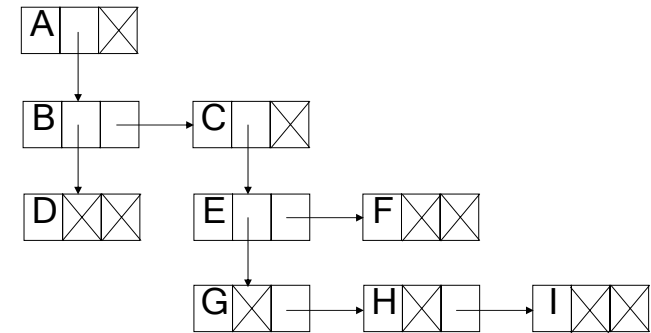
เป็นวิธีแก้ปัญหาเพื่อลดการ สลับเปลี่ยนเนื้อหาในหน่วยความจำก็คือ กำหนดลิงค์ฟิลด์ให้มีจำนวนน้อยที่สุดเท่าที่จำเป็นเท่านั้น

โดยกำหนดให้แต่ละโหนดมีจำนวนลิงค์ฟิลด์สองลิงค์ฟิลด์

-ลิงค์ฟิลด์แรกเก็บที่อยู่ของโหนดลูกคนโต

-ลิงค์ฟิลด์ที่สองเก็บที่อยู่ของโหนดพี่น้องที่เป็นโหนดถัดไป โหนดใดไม่มีโหนดลูกหรือไม่มีโหนดพี่น้องให้ค่าพอยน์เตอร์ในลิงค์ฟิลด์มีค่าเป็น Null

Tree (Cont.)

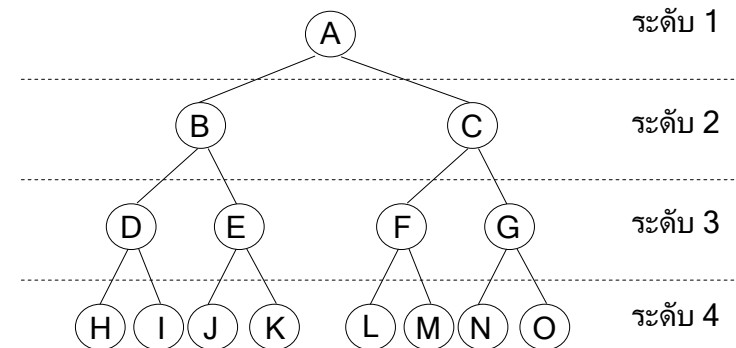


แสดงการแทนที่ทรีด้วยไบนารีทรี

Tree (Cont.)

โครงสร้างทรีที่แต่ละโหนดมีลิงค์ฟิลด์แค่สองลิงค์ฟิลด์ ซึ่งช่วยให้ประหยัดเนื้อหาในการจัดเก็บได้มาก เรียกโครงสร้างทรีที่แต่ละโหนดมีจำนวนโหนดลูกไม่เกินสองหรือแต่ละโหนดมีจำนวนทรีย่อยไม่เกินสองนี้ว่า **ไบนารีทรี** (Binary Tree)

Tree (Cont.)



แสดงไบนารีทรีแบบสมบูรณ์

Tree (Cont.)

ไบนารีทรีที่ทุก ๆ โหนดมีทริ้อยู่ทางซ้ายและ
ทริ้อยู่ทางขวา ยกเว้นโหนดใบ และโหนดใบทุกโหนด
จะต้องอยู่ที่ระดับเดียวกัน
เรียกว่า ไบนารีทรีแบบสมบูรณ์ (complete binary tree)
สามารถคำนวณจำนวนโหนดทั้งหมดใน
ไบนารีทรีแบบสมบูรณ์ได้
ถ้ากำหนดให้ L คือระดับของโหนดใด ๆ และ
N คือจำนวนโหนดทั้งหมดในทรีจะได้ว่า
ระดับ 1 มีจำนวนโหนด 1 โหนด
ระดับ 2 มีจำนวนโหนด 3 โหนด
ระดับ 3 มีจำนวนโหนด 7 โหนด

Tree (Cont.)

ระดับ L มีจำนวนโหนด $2^L - 1$ โหนด

นั่นคือ จำนวนโหนดทั้งหมดในทรีสมบูรณ์ที่มี
L ระดับ สามารถคำนวณได้จากสูตรดังนี้

$$N = 2^L - 1$$

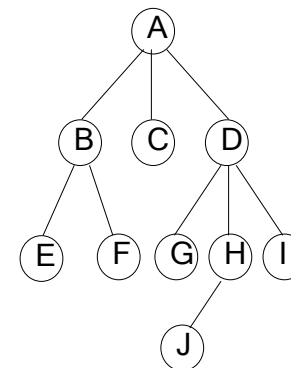
Tree (Cont.)

การแปลงทรีทั่วไปให้เป็นไบนารีทรี

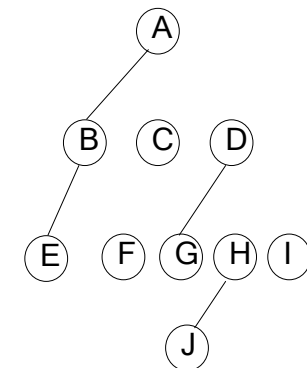
ขั้นตอนการแปลงทรีทั่วไปให้เป็น
ไบนารีทรี มีลำดับขั้นตอนการแปลง ดังต่อไปนี้

1. ให้โหนดแม่ชี้ไปยังโหนดลูกคนโต แล้วลบ
ความสัมพันธ์ ระหว่างโหนดแม่และโหนดลูกอื่น ๆ
2. ให้เชื่อมความสัมพันธ์ระหว่างโหนดพี่น้อง
3. จับให้ทริ้อยู่ทางขวาเอียงลงมา 45 องศา

Tree (Cont.)

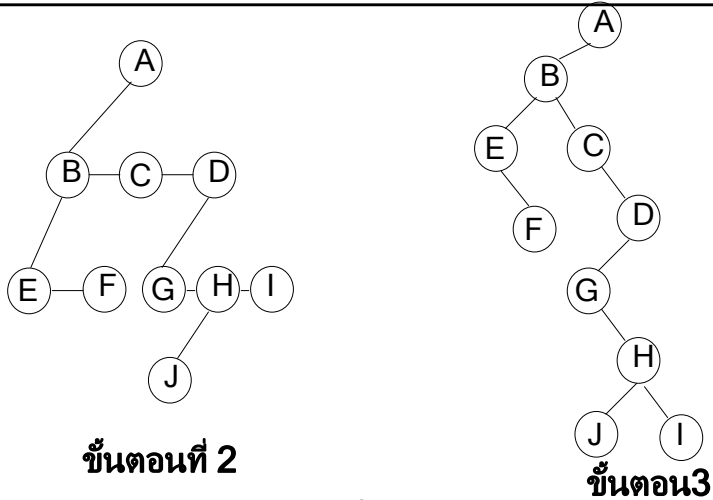


ทรีทั่วไป



ขั้นตอนที่ 1

Tree (Cont.)



Tree (Cont.)

การท่องเที่ยวในไบนารีทรี

ปฏิบัติการที่สำคัญในไบนารีทรี คือ การท่องเที่ยวในไบนารีทรี (Traversing Binary Tree) เพื่อเข้าไปเยือนทุก ๆ โหนดในทรี ซึ่งวิธีการท่องเที่ยวเข้าไปต้องเป็นไปอย่างมีระบบแบบแผน สามารถเยือนโหนดทุก ๆ โหนด ๆ ละหนึ่งครั้ง วิธีการท่องป่านั้นมีด้วยกันหลายแบบแล้วแต่ว่าต้องการลำดับขั้นตอนการเยือนอย่างไร โหนดที่ถูกเยือนอาจเป็นโหนดแม่ (แทนด้วย N) หรือโหนดลูกชาย (แทนด้วย L) หรือโหนดลูกสาว (แทนด้วย R)

Tree (Cont.)

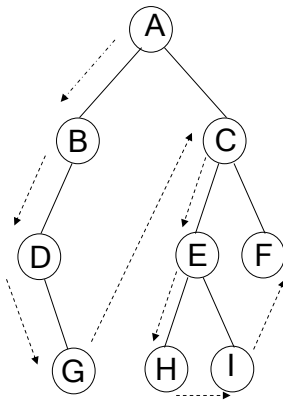
มีวิธีการท่องเที่ยวเข้าไปในทรี 6 วิธี คือ NLR LNR LRN NRL RNL และ RLN แต่วิธีการท่องเที่ยวเข้าไปในไบนารีทรีที่นิยมใช้กันมากเป็นการท่องเที่ยวจากซ้ายไปขวา 3 แบบแรกเท่านั้นคือ NLR LNR และ LRN ซึ่งลักษณะการนิยามเป็นนิยามแบบ รีเคอร์ซีฟ (Recursive) ซึ่งขั้นตอนการท่องไปในแต่ละแบบมีดังนี้

Tree (Cont.)

1. การท่องเที่ยวแบบพรีออร์เดอร์ (Preorder Traversal) เป็นการเดินเข้าไปเยือนโหนดต่าง ๆ ในทรีด้วยวิธี NLR มีขั้นตอนการเดินดังต่อไปนี้
 - (1) เยือนโหนดราก
 - (2) ท่องไปในทรีย่อยทางซ้ายแบบพรีออร์เดอร์
 - (3) ท่องไปในทรีย่อยทางขวาแบบพรีออร์เดอร์

Tree (Cont.)

NLR



เส้นทางการท่องในทรีแบบพรีออร์เดอร์ จะได้ **ABDGC**EHIF

Tree (Cont.)

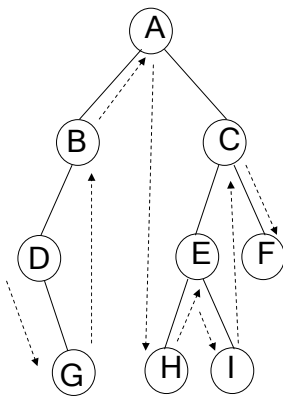
2. การท่องไปแบบอินออร์เดอร์
(Inorder Traversal)
เป็นการเดินเข้าไปเยือนโหนดต่าง ๆ
ในทรีด้วยวิธี LNR

มีขั้นตอนการเดินดังต่อไปนี้

- (1) ท่องไปในทรีย่อยทางซ้ายแบบอินออร์เดอร์
- (2) เยือนโหนดราก
- (3) ท่องไปในทรีย่อยทางขวาแบบอินออร์เดอร์

Tree (Cont.)

LNR



เส้นทางการท่องในทรีแบบอินออร์เดอร์ จะได้ **DGBA**HEICF

Tree (Cont.)

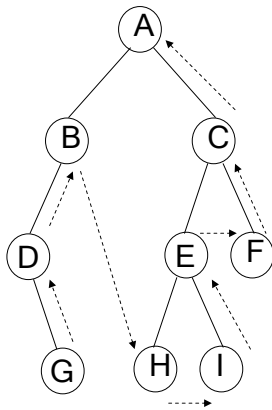
3. การท่องไปแบบโพสต์ออร์เดอร์
(Postorder Traversal)

เป็นการเดินเข้าไปเยือนโหนดต่าง ๆ
ในทรีด้วยวิธี LRN มีขั้นตอนการเดินดังต่อไปนี้

- (1) ท่องไปในทรีย่อยทางซ้ายแบบโพสต์ออร์เดอร์
- (2) ท่องไปในทรีย่อยทางขวาแบบโพสต์ออร์เดอร์
- (3) เยือนโหนดราก

Tree (Cont.)

LRN



เส้นทางการท่องในทรีแบบโพสต์ออร์เดอร์ จะได้ **GDBHIEFCA**

Tree (Cont.)

เอ็กซ์เพรสชันทรี (Expression Tree)

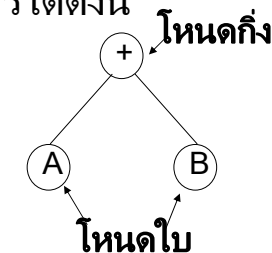
เป็นการนำเอาโครงสร้างทรีไปใช้เก็บนิพจน์ทางคณิตศาสตร์โดยเป็นไบนารีทรี ซึ่งแต่ละโหนดเก็บตัวดำเนินการ (Operator) และและตัวถูกดำเนินการ (Operand) ของนิพจน์คณิตศาสตร์นั้น ๆ ไว้ หรืออาจจะเก็บค่านิพจน์ทางตรรกะ (Logical Expression) นิพจน์เหล่านี้เมื่อแทนในทรีต้องคำนึงลำดับขั้นตอนในการคำนวณตามความสำคัญของเครื่องหมายด้วยโดยมีความสำคัญตามลำดับดังนี้

Tree (Cont.)

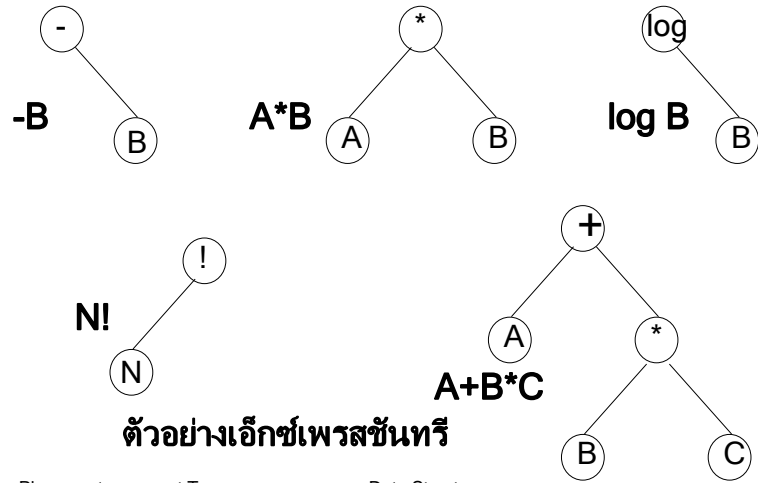
- ฟังก์ชัน
- วงเล็บ
- ยกกำลัง
- เครื่องหมายหน้าเลขจำนวน (unary)
- คูณ หรือหาร
- บวก หรือ ลบ
- ถ้ามีเครื่องหมายที่ระดับเดียวกันให้ทำจากซ้ายไปขวา

Tree (Cont.)

การแทนนิพจน์ในเอ็กซ์เพรสชันทรี ตัวถูกดำเนินการจะเก็บอยู่ที่โหนดใบ ส่วนตัวดำเนินการจะเก็บในโหนดกิ่ง หรือโหนดที่ไม่ใช่โหนดใบ เช่น นิพจน์ $A + B$ สามารถแทนในเอ็กซ์เพรสชันทรีได้ดังนี้



Tree (Cont.)



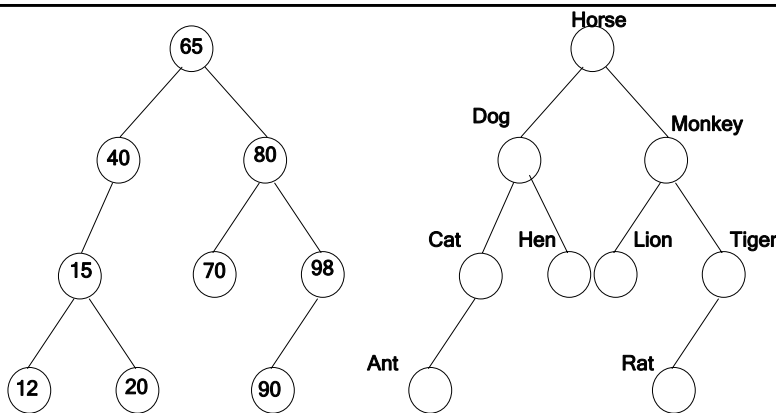
ตัวอย่างเอ็กซ์เพรสชันทรี

Tree (Cont.)

ไบนารีเซิร์ชทรี

ไบนารีเซิร์ชทรี (Binary Search Tree) เป็นไบนารีทรีที่มีคุณสมบัติที่ว่าทุก ๆ โหนดในทรี ค่าของโหนดรากมีค่ามากกว่าค่าของทุกโหนดในทรีย่อยทางซ้าย และมีค่าน้อยกว่าหรือเท่ากับค่าของทุกโหนดในทรีย่อยทางขวา และในแต่ละทรีย่อยก็มี คุณสมบัติเช่นเดียวกัน

Tree (Cont.)



ไบนารีเซิร์ชทรีของเลขจำนวน ไบนารีเซิร์ชทรีของชื่อสัตว์

Tree (Cont.)

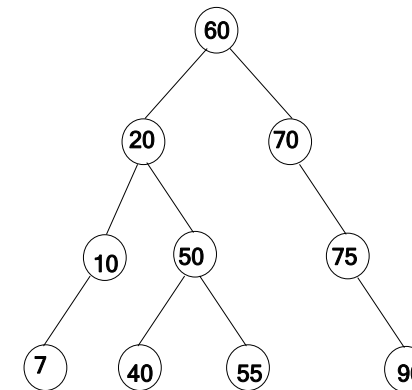
ปฏิบัติการในไบนารีเซิร์ชทรี ปฏิบัติการเพิ่มโหนดเข้าหรือดึงโหนดออกจากไบนารีเซิร์ชทรีค่อนข้างยุ่งยากกว่าปฏิบัติการในโครงสร้างอื่น ๆ เนื่องจากหลังปฏิบัติการเสร็จเรียบร้อยแล้วต้องคำนึงถึงความเป็นไบนารีเซิร์ชทรีของทรีนั้นด้วยซึ่งมีปฏิบัติการดังต่อไปนี้

Tree (Cont.)

(1) การเพิ่มโหนดในไบนารีเซิร์ชทรี การเพิ่มโหนดใหม่เข้าไปในไบนารีเซิร์ชทรี ถ้าทรีว่างโหนดที่เพิ่มเข้าไปก็จะเป็นโหนดรากของทรี ถ้าทรีไม่ว่างต้องทำการตรวจสอบว่าโหนดใหม่ที่เพิ่มเข้ามานั้นมีค่ามากกว่าหรือน้อยกว่าค่าที่โหนดราก ถ้ามีค่ามากกว่าหรือเท่ากันจะนำโหนดใหม่ไปเพิ่มในทรีย่อยทางขวา และถ้ามีค่าน้อยกว่านำโหนดใหม่ไปเพิ่มในทรีย่อยทางซ้าย ในทรีย่อยนั้นต้องทำการเปรียบเทียบในลักษณะเดียวกันจนกระทั่งหาตำแหน่งที่สามารถเพิ่มโหนดได้ ซึ่งโหนดใหม่ที่เพิ่มในทรีในที่สุดจะต้องเป็นโหนดใบ

Tree (Cont.)

จากตัวเลขจำนวนเต็มต่อไปนี้ เมื่อนำไปสร้างเป็นไบนารีจะทำได้ดังรูป 60 , 20 , 70 , 75 , 50 , 40 , 10 , 90 , 7 , 55



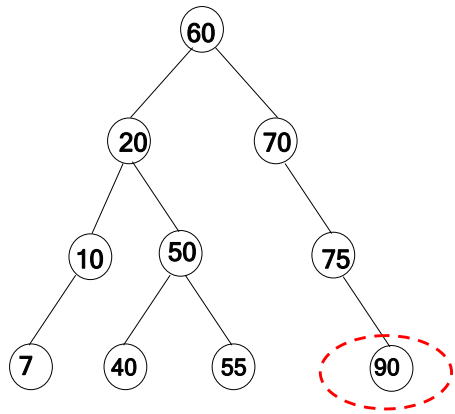
Tree (Cont.)

(2) การดึงโหนดในไบนารีเซิร์ชทรี หลังจากดึงโหนดที่ต้องการออกจากทรีแล้ว ทรีนั้นต้องคงสภาพไบนารีเซิร์ชทรีเหมือนเดิมก่อนที่จะทำการดึงโหนดใด ๆ ออกจากไบนารีเซิร์ชทรี ต้องค้นหาก่อนว่าโหนดที่ต้องการดึงออกอยู่ที่ตำแหน่งไหนภายในทรี และต้องทราบที่อยู่ของโหนดแม่โหนดนั้นด้วย แล้วจึงทำการดึงโหนดออกจากทรีได้ ขั้นตอนวิธีดึงโหนดออกอาจแยกพิจารณาได้ 3 กรณีดังต่อไปนี้

Tree (Cont.)

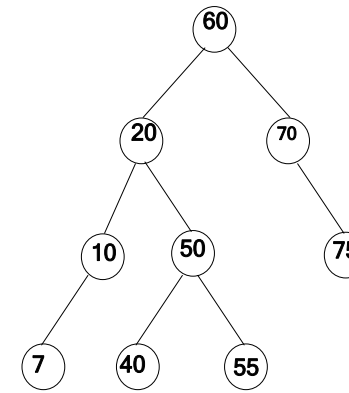
ก. กรณีโหนดที่จะดึงออกเป็นโหนดใบ การดึงโหนดใบออกในกรณีนี้ทำได้ง่ายที่สุด โดยการดึงโหนดนั้นออกได้ทันที เนื่องจากไม่กระทบกับโหนดอื่นมากนัก วิธีการก็คือให้ค่าในลิงค์ฟิลด์ของโหนดแม่ซึ่งเก็บที่อยู่ของโหนดที่ต้องการดึงออกให้มีค่าเป็น Null

Tree (Cont.)



โหนดที่ต้องการดึงออก

Tree (Cont.)

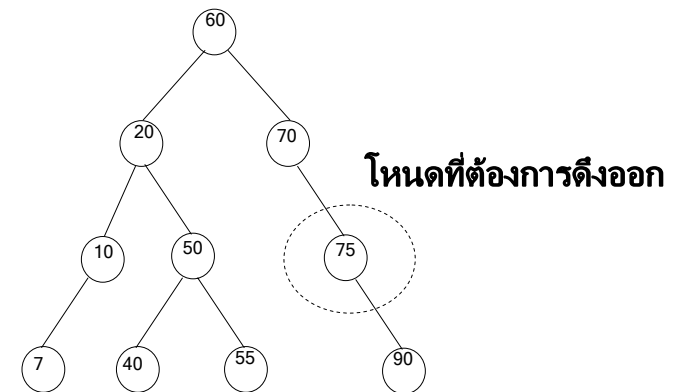


แสดงไบนารีเซิร์ชทรีหลังดึงโหนดออกแล้ว

Tree (Cont.)

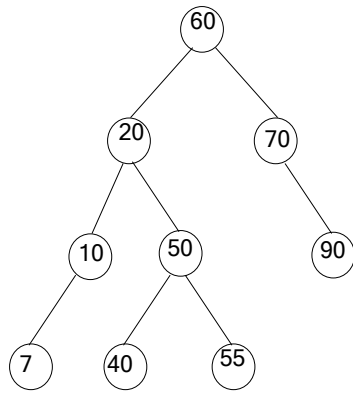
ข. กรณีโหนดที่ดึงออกมีเฉพาะ
ทริ้อย่างซ้ายหรือทริ้อย่างขวาเพียง
ด้านใดด้านหนึ่ง วิธีการดึงโหนดนี้ออก
สามารถใช้วิธีการเดียวกับการดึงโหนด
ออกจากลิงค์ลิสต์ โดยให้โหนดแม่ของ
โหนดที่จะดึงออกชี้ไปยังโหนดลูกของ
โหนดนั้นแทน

Tree (Cont.)



โหนดที่ต้องการดึงออก

Tree (Cont.)



แสดงไบนารีเซิร์ชทรีหลังดึงโหนดออกแล้ว

Tree (Cont.)

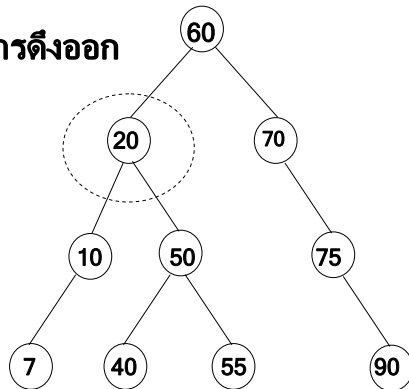
ค. กรณีโหนดที่ดึงออกมีทั้งทริ้อยทางซ้ายและทริ้อยทางขวาต้องเลือกโหนดมาแทนโหนดที่ถูกดึงออก โดยอาจจะเลือกมาจากทริ้อยทางซ้ายหรือทริ้อยทางขวาก็ได้

- ถ้าโหนดที่มาแทนที่เป็นโหนดที่เลือกจากทริ้อยทางซ้ายต้องเลือกโหนดที่มีค่ามากที่สุด ในทริ้อยทางซ้ายนั้น

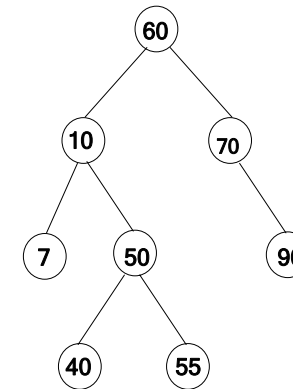
- ถ้าโหนดที่จะมาแทนที่เป็นโหนดที่เลือกมาจากทริ้อยทางขวา ต้องเลือกโหนดที่มีค่าน้อยที่สุดในทริ้อยทางขวานั้น

Tree (Cont.)

โหนดที่ต้องการดึงออก

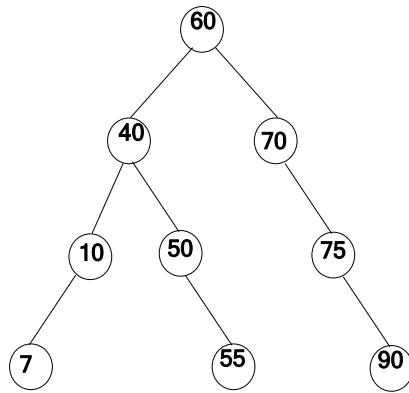


Tree (Cont.)



แสดงไบนารีเซิร์ชทรีหลังดึงโหนดออกแล้ว กรณีใช้ทริ้อยทางซ้ายมาแทน

Tree (Cont.)



แสดง ไบนารีเซิร์ชทรีหลังดึงโหนดออกแล้ว กรณีใช้ทรีย่อยทางขวามาแทน

แบบฝึกหัด

1. ให้นักศึกษาอธิบายคำที่เกี่ยวข้องกับ ทรี
ดังนี้ Parent , Child , Root , Sibling
2. ไบนารีทรี คืออะไร มีหลักในการทำงาน
อย่างไร ได้แก่อะไรที่มีลักษณะอย่างไรบ้าง
3. จากตัวเลขต่อไปนี้ ให้นักศึกษาดำเนินการให้
เป็น ไบนารีทรี แบบสมบูรณ์
10,5,8,7,15,20,28,9,12,6
พร้อมทำการท่องไปในทรีแบบ NLR , LRN