



มหาวิทยาลัยราชภัฏนครปฐม



## บทที่ 7 ต้นไม้

ผู้บรรยาย : ผศ.ดร.ณัฐชามณูห์ ศรีจำเริญรัตน์  
สาขาวิชาคอมพิวเตอร์ธุรกิจ คณะวิทยาการจัดการ  
มหาวิทยาลัยราชภัฏนครปฐม



มหาวิทยาลัยราชภัฏนครปฐม  
Nakhon Pathom Rajabhat University



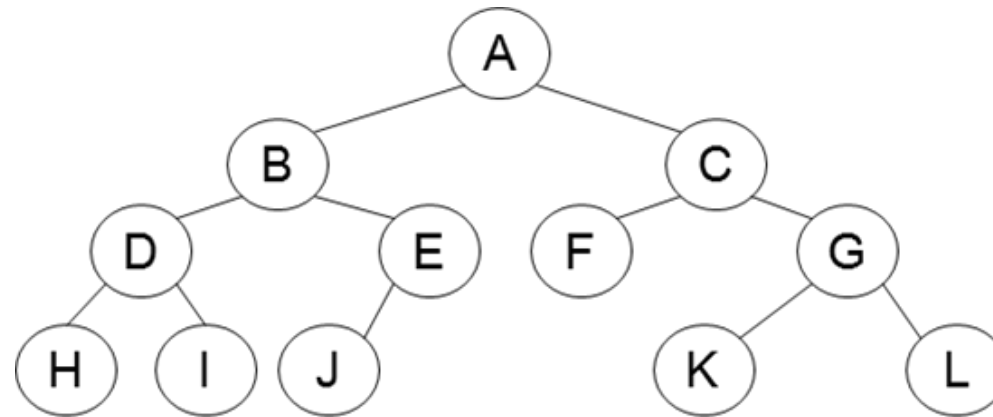
# ต้นไม้

- การเก็บข้อมูลโครงสร้างแบบต้นไม้ (tree) ข้อมูลจะถูกเก็บในลักษณะความสัมพันธ์แบบลำดับชั้น เช่น แผนผังการบริหารงานในหน่วยงาน การเขียนโครงสร้างองค์กร
- ความสัมพันธ์ของข้อมูล โครงสร้างต้นไม้ (tree) มีลักษณะคล้ายกิ่งก้านของต้นไม้ ดังนี้
  - ต้นไม้ในตามธรรมชาตินั้นจะเจริญเติบโตจากด้านล่างพื้นดินไปบนอากาศ
  - ส่วนโครงสร้างข้อมูลลักษณะต้นไม้ นั้นจะกำหนดให้เจริญเติบโตจากด้านบนลงมาล่าง
- โครงสร้างข้อมูลแบบต้นไม้จะมีลักษณะคล้ายกับต้นไม้ในธรรมชาติ กล่าวคือต้นไม้มีราก (root) และแต่ละโหนดจะสัมพันธ์กันเหมือนครอบครัว โดยมีการกำหนดโหนดพ่อแม่เป็นโหนดที่อยู่ก่อนโหนดที่กำลังพิจารณา และโหนดลูกเป็นโหนดที่อยู่หลังโหนดที่กำลังพิจารณา ส่วนโหนดใบ (leaf) จะเป็นโหนดที่อยู่ลำดับล่างสุดของต้นไม้ นั่นเองเปรียบได้กับใบของต้นไม้ที่จะอยู่ด้านนอกสุดของต้นไม้ นั่นเอง

## 7.1 แนวคิดพื้นฐานของต้นไม้(ต่อ)

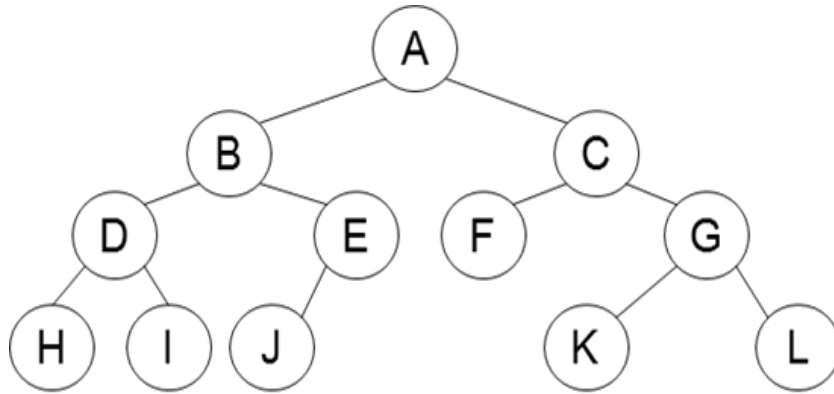
### องค์ประกอบของต้นไม้

เนื่องจากต้นไม้มีโครงสร้างที่มีลำดับคล้ายชั้นลำดับบรรพบุรุษ จึงมีการเรียกส่วนต่าง ๆ ของต้นไม้คล้ายกับการเรียกความสัมพันธ์ระหว่างแม่ลูก



## 7.1 แนวคิดพื้นฐานของต้นไม้(ต่อ)

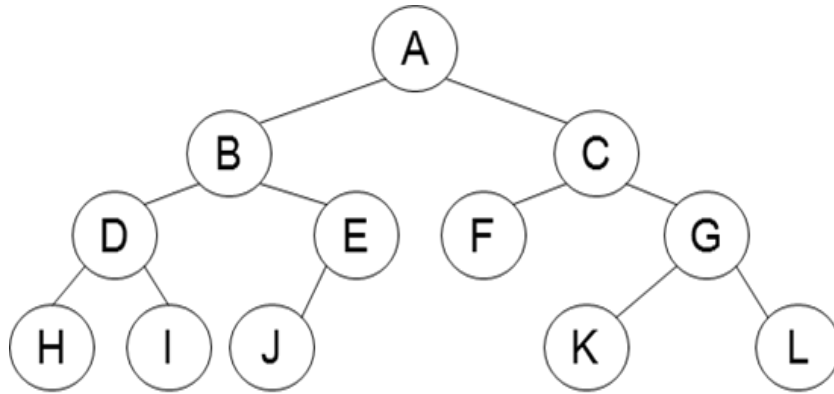
### องค์ประกอบของต้นไม้



- โหนดราก หรือ รุทโหนด (root node) เป็นโหนดที่อยู่สูงสุดในโครงสร้างต้นไม้
  - คือ โหนด A
- โหนดพ่อแม่ (parent node) เป็นโหนดที่อยู่สูงกว่าโหนดที่กำลังพิจารณาเชื่อมต่อ
  - คือ โหนด A, B, C, D, E, G
- โหนดลูก (child node) โหนดที่อยู่ต่ำกว่าโหนดที่กำลังพิจารณาและมีเส้นทางทำการเชื่อมต่ออยู่
  - คือ โหนด B, C, D, E, F, G, H, I, J, K, L

## 7.1 แนวคิดพื้นฐานของต้นไม้(ต่อ)

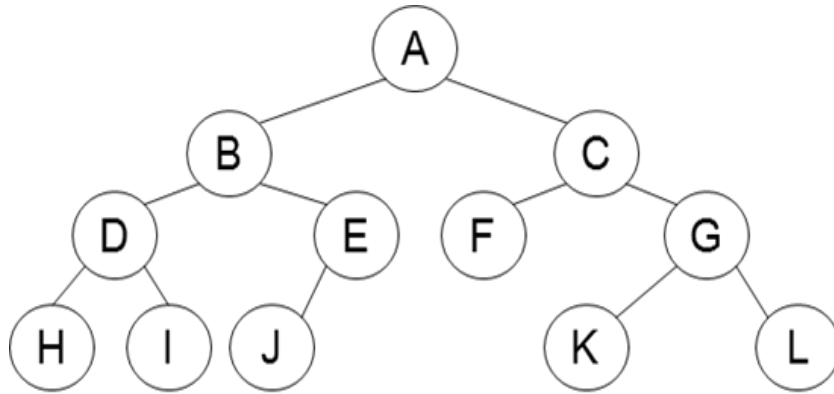
### องค์ประกอบของต้นไม้



- โหนดพี่น้อง (sibling node) เป็นโหนดที่มีโหนดพ่อแม่เดียวกัน
  - คือ โหนด {B,C}, {D, E}, {F, G}, {H,I}, {J}, {K,L}
- โหนดใบ (leaf node) เป็นโหนดที่ไม่มีเส้นทางเชื่อมต่อไปยังโหนดอื่น ๆ หรือโหนดลูกนั่นเอง
  - คือ โหนด H, I, J, F, K, L
- ดีกรี (degree) เป็นจำนวนโหนดลูกของโหนดที่กำลังพิจารณา
  - เช่น โหนด D มีดีกรีเท่ากับ 2

## 7.1 แนวคิดพื้นฐานของต้นไม้(ต่อ)

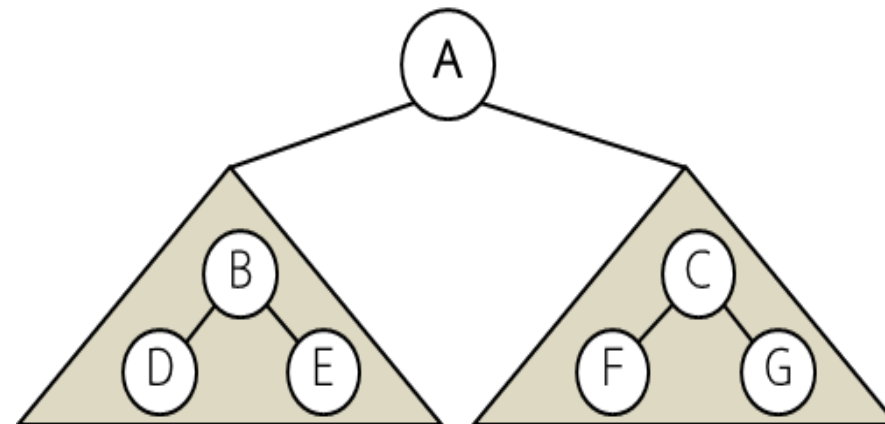
### องค์ประกอบของต้นไม้



- เส้นทาง (path) เป็นเส้นเชื่อมระหว่างโหนดต้นทางไปยังโหนดที่ต้องการ
  - คือ โหนด A-B-D-H
- ความลึกของโหนด (depth) เป็นความยาวของ path ที่นับจากโหนดราก ไปยังโหนดที่ต้องการโดย รากโหนดจะมีความลึกเท่ากับ 0
- ความสูงของต้นไม้ (height) เป็นระยะทางที่ยาวที่สุดจากรากโหนดมายังโหนดใบ

## 7.2 ต้นไม้ทวิภาค


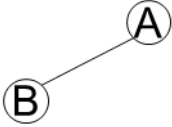
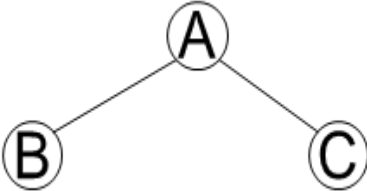
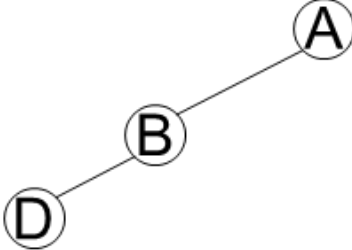
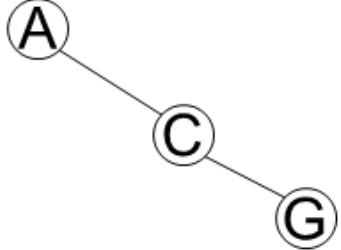
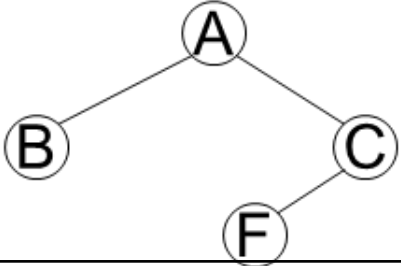
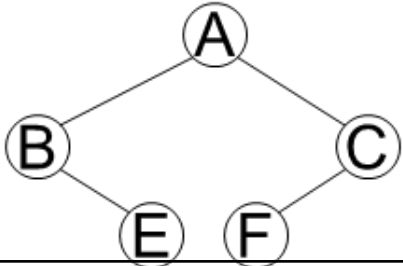
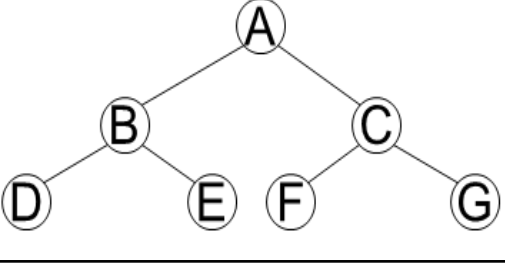
ต้นไม้ทวิภาค (binary tree) เป็นต้นไม้ที่มีคุณสมบัติเด่นที่แต่ละโหนดมีโหนดใบได้ไม่เกิน 2 โหนด และโหนดลูกจะแบ่งเป็นโหนดลูกที่อยู่ด้านซ้าย (left child) และโหนดลูกที่อยู่ด้านขวา (right child) หรือแต่ละเลเวลมีโหนด 2 โหนดครบทั้งด้านซ้ายและด้านขวา จะเรียกว่าต้นไม้สมบูรณ์ (perfect binary tree)





## 7.2 ต้นไม้ทวิภาค (ต่อ)

โครงสร้างต้นไม้ทวิภาครูปแบบต่าง ๆ

No nodes		
(a)	(b)	(c)
		
(d)	(e)	(f)
		
(g)	(h)	(i)

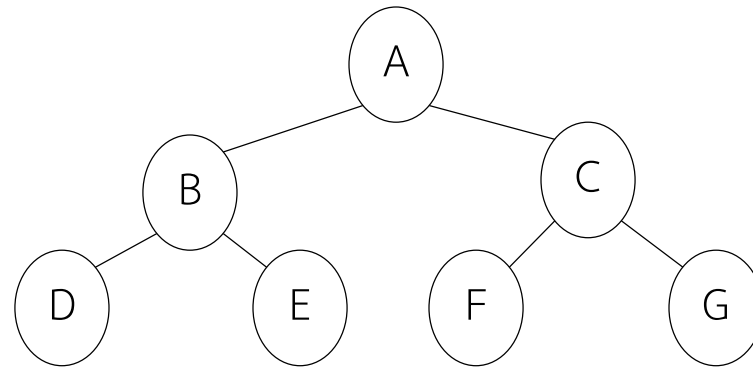


## 7.3 การแทนต้นไม้ทวิภาคในหน่วยความจำ

- สามารถทำการแทนหน่วยความจำด้วยอาร์เรย์หนึ่งมิติซึ่งสามารถใช้ต้นไม้ทวิภาคแบบสมบูรณ์ในการแทนโครงสร้างต้นไม้ ซึ่งการแทนนั้นจะต้องแทนทีละลำดับชั้นข้อมูลจากซ้ายไปขวาตามลำดับการแทนข้อมูล ดังนี้
  1. Root node จะกำหนดให้แทนอาร์เรย์ ตำแหน่งที่ 1
  2. โหนด ลำดับถัดมาจะแทนข้อมูลในอาร์เรย์ตามลำดับ level ที่ระดับ level ลงในอาร์เรย์จนครบทุกโหนด
  3. กรณี ต้นไม้เป็นต้นไม้ไม่สมบูรณ์หรือเกือบสมบูรณ์ให้ทำการเพิ่มโหนดว่างลงในต้นไม้ให้เต็มก่อนการแทนต้นไม้ทวิภาคในหน่วยความจำ

### 7.3 การแทนต้นไม้ทวิภาคในหน่วยความจำ(ต่อ)

การแทนต้นไม้ทวิภาคสมบูรณ์ในอาร์เรย์หนึ่งมิติ

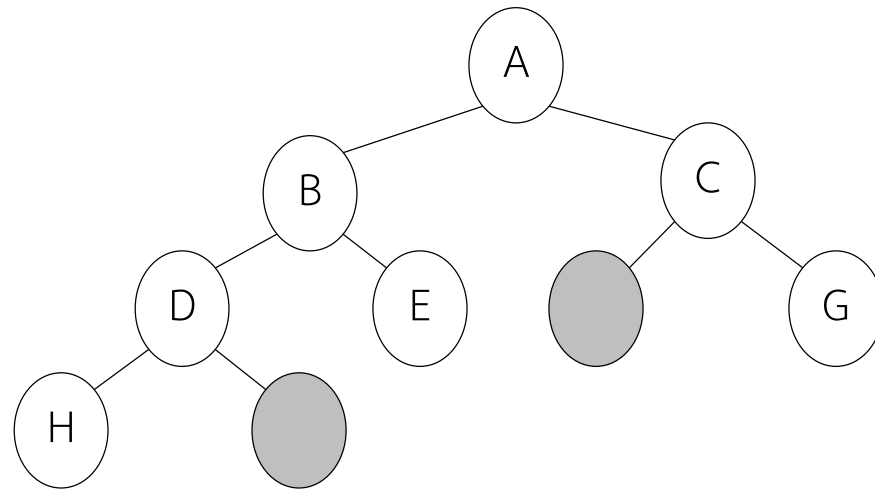


[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
	A	B	C	D	E	F	G



### 7.3 การแทนต้นไม้ทวิภาคในหน่วยความจำ(ต่อ)

การแทนต้นไม้ทวิภาคแบบไม่สมบูรณ์หรือเกือบสมบูรณ์ในอาร์เรย์หนึ่งมิติ



[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
	A	B	C	D	E		G	H	



## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค

- การท่องเข้าไปในต้นไม้แบบทวิภาค (binary tree traversals)

โดยปกติจะสามารถทำได้ 2 วิธีด้วยกันคือ

- วิธีการท่องแบบแนวลึก (depth-first traversals)
- วิธีการท่องแบบแนวกว้าง (breadth first traversals)



## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

### 7.4.1 วิธีการท่องแบบแนวลึก

การท่องแบบแนวลึก (depth-first traversals) เริ่มจากกำหนดโหนดเริ่มต้นจากนั้นทำการค้นหาโหนดที่เชื่อมกับโหนดเริ่มต้น จากนั้นจะทำการค้นหาต่อไปเรื่อย ๆ ตามลำดับ โดยวิธีการท่องแบบแนวลึก สามารถแบ่งย่อยได้ 3 วิธี ดังต่อไปนี้

- แบบ preorder traversal
- แบบ inorder traversal
- แบบ postorder traversal



## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

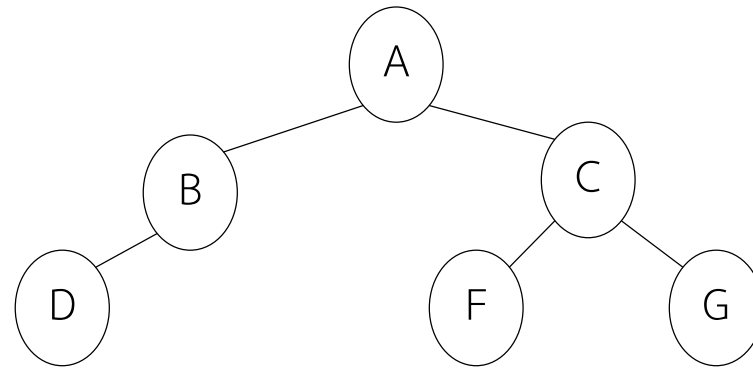
### 1) แบบ preorder traversal (NLR)

การท่องไปในต้นไม้แบบ preorder มีลำดับวิธีการท่องเพื่อค้นหาข้อมูลในการเข้าถึงแต่ละโหนด ดังต่อไปนี้

1. เริ่มต้นจากรูทโหนด
2. ท่องเข้าไปในซ้บตรีด้านซ้าย
3. ท่องเข้าไปในซ้บตรีด้านขวา เป็นลำดับสุดท้าย

## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

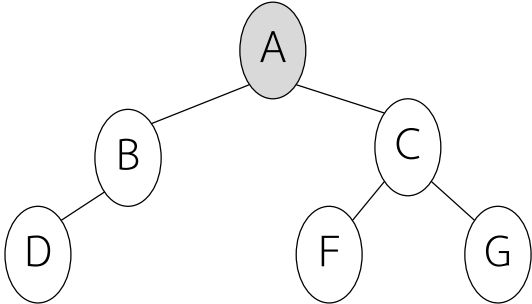
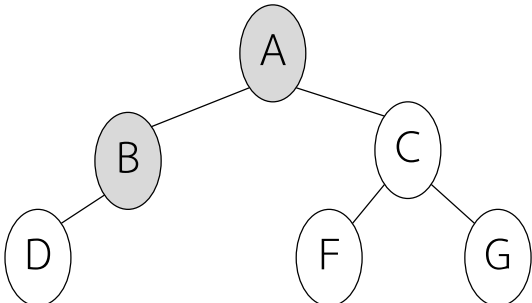
ตัวอย่างที่ 7.1 การแสดงขั้นตอนการท่องไปในต้นไม้แบบ preorder





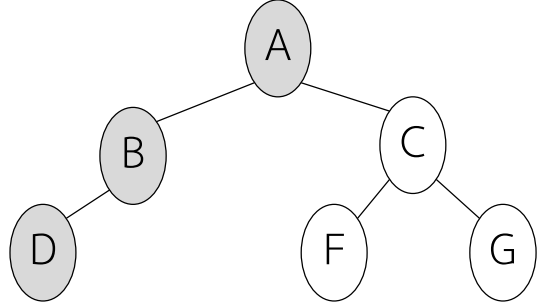
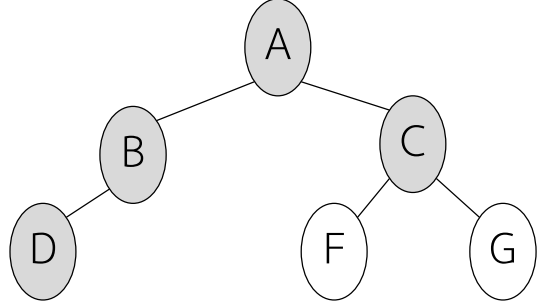
## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.1 การแสดงขั้นตอนการท่องไปในต้นไม้แบบ preorder

ลำดับ	การดำเนินการ	ผลลัพธ์	รูปภาพ
1	เยี่ยมรูทโหนด	A	
2	ท่องเข้าไปในซับทรีด้านซ้ายของทรีย่อย B	B	

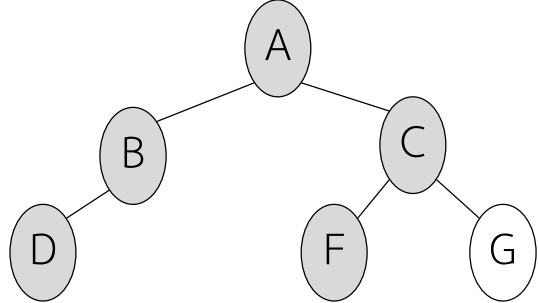
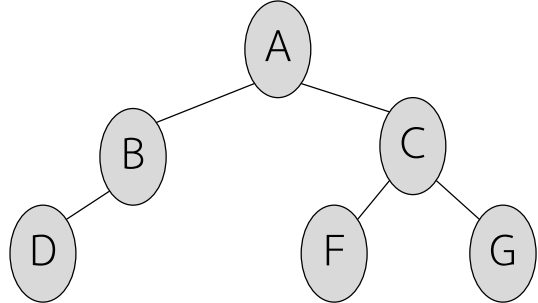
## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.1 การแสดงขั้นตอนการท่องไปในต้นไม้แบบ preorder

ลำดับ	การดำเนินการ	ผลลัพธ์	รูปภาพ
3	ท่องเข้าไปในซ้บตรีด้านซ้ายของโหนด D	D	
เข้าถึงโหนดซ้บตรีด้านซ้ายครบแล้ว			
4	ท่องเข้าไปในซ้บตรีด้านขวาของทริย่อย C	C	

## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

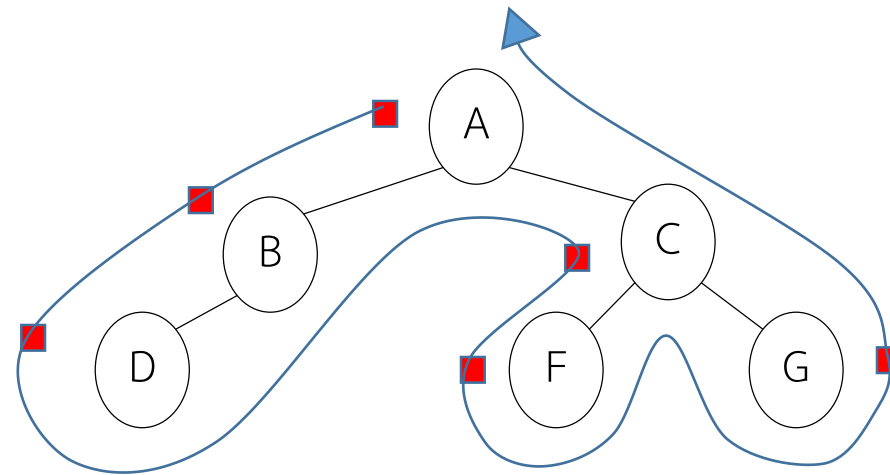
ตัวอย่างที่ 7.1 การแสดงขั้นตอนการท่องไปในต้นไม้แบบ preorder

ลำดับ	การดำเนินการ	ผลลัพธ์	รูปภาพ
5	ท่องเข้าไปในซีกทางด้านซ้ายของโหนด F	F	
6	ท่องเข้าไปในซีกด้านขวาของโหนด G	G	
จบการทำงาน			

## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ผลลัพธ์ของการท่องเข้าไปในทรีแบบ preorder

สามารถเข้าถึงข้อมูลแต่ละโหนด ดังนี้ A B D C F G ดังนี้





## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

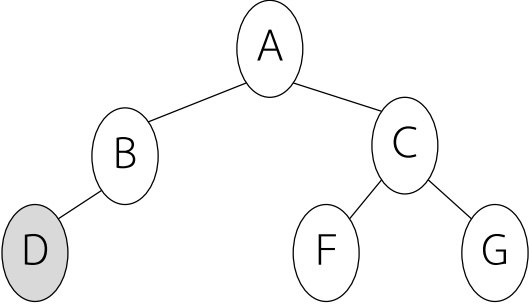
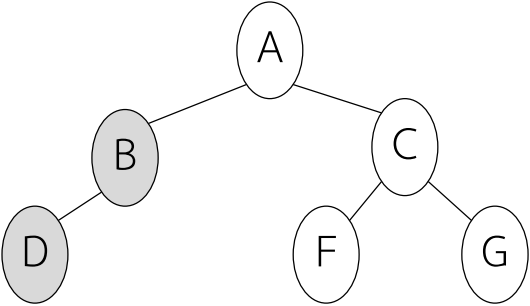
### 2) แบบ inorder traversal (LNR)

การท่องไปในต้นไม้แบบ inorder มีลำดับวิธีการท่องเพื่อค้นหาข้อมูลในการเข้าถึงแต่ละโหนด ดังต่อไปนี้

- 1) เริ่มต้นจากท่องเข้าไปในซ้บตรีด้านซ้าย
- 2) ท่องไปที่รูทโหนด
- 3) ท่องเข้าไปในซ้บตรีด้านขวา เป็นลำดับสุดท้าย

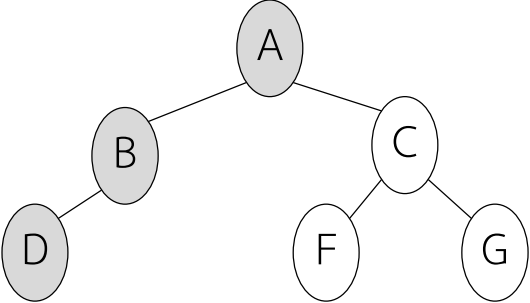
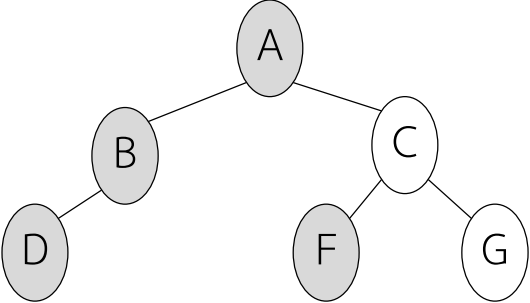
## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.2 การแสดงขั้นตอนการท่องไปในต้นไม้แบบ inorder

ลำดับ	การดำเนินการ	ผลลัพธ์	รูปภาพ
1	เริ่มต้นจากท่องเข้าไปในซับทรีด้านซ้ายของ inorder	-	-
2	ท่องเข้าไปในทรีจนถึงโหนดใบที่อยู่ซับทรีด้านซ้ายคือ โหนด D	D	
3	ท่องไปที่รูทโหนด ตอนนีรูทโหนดคือ โหนด B	B	

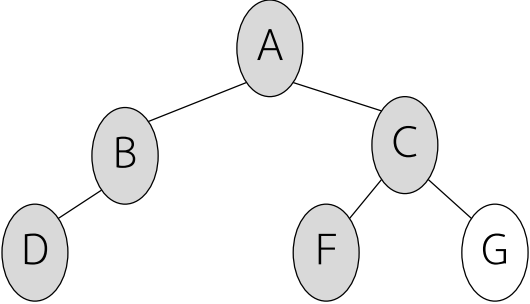
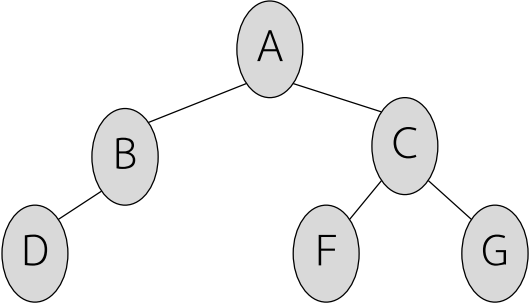
## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.2 การแสดงขั้นตอนการท่องไปในต้นไม้แบบ inorder

ลำดับ	การดำเนินการ	ผลลัพธ์	รูปภาพ
4	โหนด B เป็นซับทรีด้านซ้ายของรูทโหนด A จึงได้โหนด A เป็นลำดับถัดมา	A	
ซับทรีด้านซ้ายได้ท่องไปครบทุกโหนดและกลับมาที่รูทแล้ว ลำดับต่อไปท่องซับทรีด้านขวา			
5	โหนด C เป็นซับทรีด้านขวาของรูทโหนด A แต่เป็นรูทโหนดของซับทรีด้านซ้ายของ F จึงได้โหนด F เป็นลำดับถัดมา	F	

## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.2 การแสดงขั้นตอนการท่องไปในต้นไม้แบบ inorder

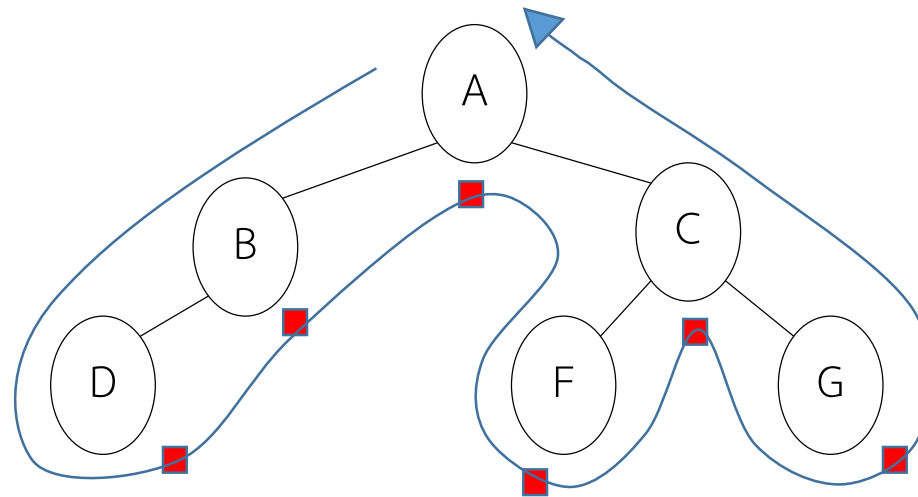
ลำดับ	การดำเนินการ	ผลลัพธ์	รูปภาพ
6	หลังจากได้โหนด F เป็นซับทรี ด้านซ้ายให้ท่องไปที่รูทโหนดซึ่งคือ โหนด C จึงได้โหนด C เป็นลำดับถัดมา	C	
7	จากรูปแบบการท่องแบบ inorder หลังจากท่องไปที่รูทโหนดแล้วให้ท่องไปที่ซับทรีด้านขวา จึงได้โหนด G เป็นลำดับถัดมา	G	



## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ผลลัพธ์ของการท่องเข้าไปในต้นไม้แบบ inorder

สามารถเข้าถึงข้อมูลแต่ละโหนด ดังนี้ D B A F C G ดังนี้





## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

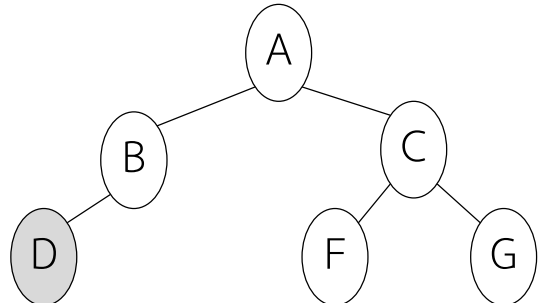
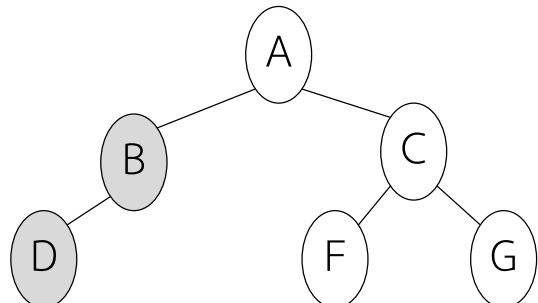
### 3) แบบ postorder traversal (LRN)

การท่องไปในต้นไม้แบบ postorder มีลำดับวิธีการท่องเพื่อค้นหาข้อมูลในการเข้าถึงแต่ละโหนด ดังต่อไปนี้

- 1) เริ่มต้นจากท่องเข้าไปในซับทรีด้านซ้าย
- 2) ท่องเข้าไปในซับทรีด้านขวา
- 3) ท่องไปที่รูทโหนด เป็นลำดับสุดท้าย

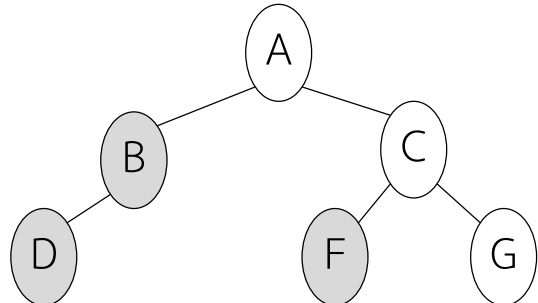
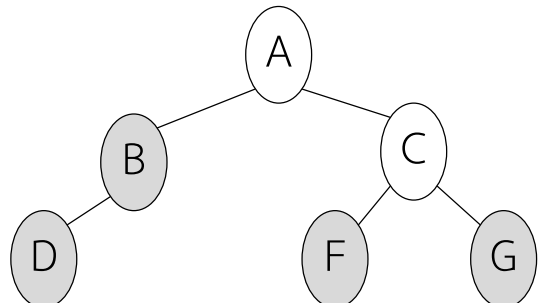
## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.3 การแสดงขั้นตอนการท่องไปในทรีแบบ postorder

ลำดับ	การดำเนินการ	ผลลัพธ์	รูปภาพ
1	เริ่มต้นจากท่องเข้าไปในซับทรีด้านซ้ายของ postorder	-	
2	ท่องเข้าไปในทรีจนถึงโหนดใบที่อยู่ซับทรีด้านซ้ายคือ โหนด D	D	
3	ขณะนี้โหนด B เป็นรูท และทางซับทรีด้านขวาไม่มีโหนด จึงได้โหนด B เป็นลำดับถัดมา	B	

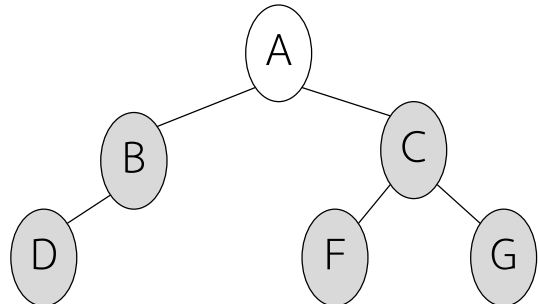
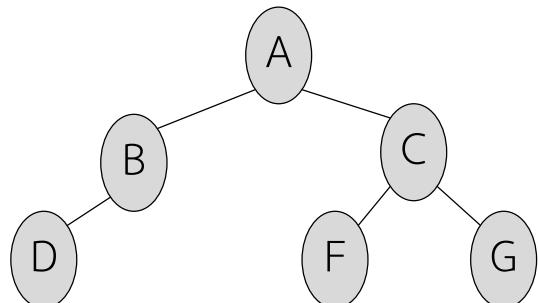
## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.3 การแสดงขั้นตอนการท่องไปในทรีแบบ postorder

ลำดับ	การดำเนินการ	ผลลัพธ์	รูปภาพ
ชั้นทรีด้านซ้ายได้ท่องไปครบทุกโหนดและกลับมาที่รูท A แล้ว ลำดับต่อไปท่องชั้นทรีด้านขวา ก่อนจะกลับมาท่องที่โหนดรูท A เป็นลำดับสุดท้าย			
4	ชั้นทรีด้านขวาของโหนด A มีโหนด C เป็นรูทโหนดของชั้นทรีด้านขวาของโหนด B จึงได้โหนด F เป็นลำดับถัดมา	F	
5	ขณะนี้โหนด C เป็นรูท และทางชั้นทรีด้านขวาคือโหนด G จึงได้โหนด G เป็นลำดับถัดมา	G	

## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

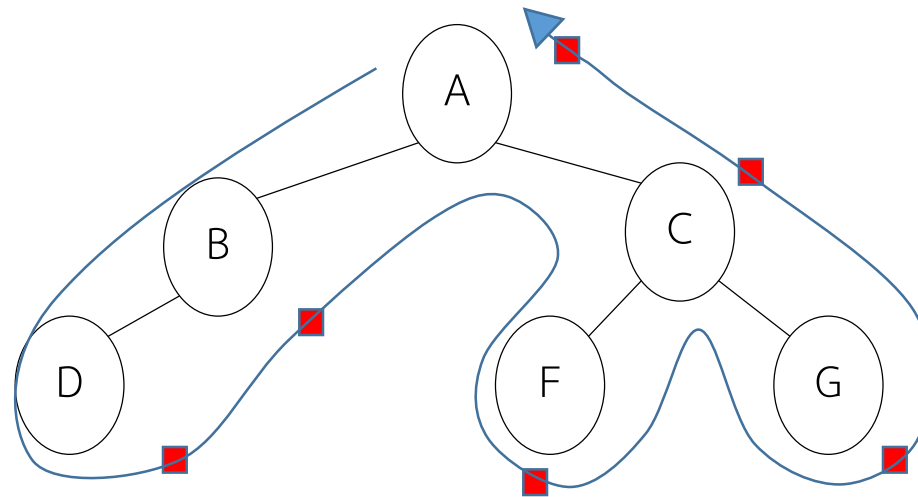
ตัวอย่างที่ 7.3 การแสดงขั้นตอนการท่องไปในทรีแบบ postorder

ลำดับ	การดำเนินการ	ผลลัพธ์	รูปภาพ
6	ขณะนี้ท่องซึบทรีด้านซ้ายและซึบทรีด้านขวา ของรูท C หมดจึงกลับมาที่รูท C จึงได้โหนด C เป็นลำดับถัดมา	C	
7	โหนด A เป็นรูท และทางซึบทรีด้านขวาของโหนด A ได้ท่องหมดแล้ว จึงได้โหนด A เป็นลำดับถัดมา	A	

## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ผลลัพธ์ของการท่องเข้าไปในต้นไม้แบบ postorder

สามารถเข้าถึงข้อมูลแต่ละโหนด ดังนี้ D B F G C A ดังนี้





## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค

แบบ preorder, inorder, postorder มีขั้นตอนดังนี้

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <iostream.h>
4
5 struct node
6 {
7     int data;
8     struct node* left;
9     struct node* right;
10 };
11
```

```
12 struct node* newNode(int data)
13 {
14     struct node* node = (struct node*)
15         malloc(sizeof(struct node));
16     node->data = data;
17     node->left = NULL;
18     node->right = NULL;
19     return(node);
20 }
21
```



## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ 1)

```
22 void Postorder(struct node* node)
23 {
24     if (node == NULL)
25         return;
26     Postorder(node->left);
27     Postorder(node->right);
28     cout<<node->data<<" ";
29 }
30
```

```
31 void Inorder(struct node* node)
32 {
33     if (node == NULL)
34         return;
35     Inorder(node->left);
36     cout<<node->data<<" ";
37     Inorder(node->right);
38 }
39
```





## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ 2)

```
40 void Preorder(struct node* node)
41 {
42     if (node == NULL)
43         return;
44     cout<<node->data<<" ";
45     Preorder(node->left);
46     Preorder(node->right);
47 }
48
```

```
49 int main()
50 {
51     struct node *root = newNode(1);
52     root->left      = newNode(2);
53     root->right     = newNode(3);
54     root->left->left = newNode(4);
55     root->left->right = newNode(5);
56
```



## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ 3)

```
57 cout<<"\n Preorder traversal of binary tree is \n";
58     Preorder(root);
59     cout<<"\n Inorder traversal of binary tree is \n";
60     Inorder(root);
61     cout<<"\n Postorder traversal of binary tree is \n";
62     Postorder(root);
63     system("pause");
64     return 0;
65 }
```



## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

### ผลการทำงานของโปรแกรม

Preorder traversal

1 2 4 5 3

Inorder traversal

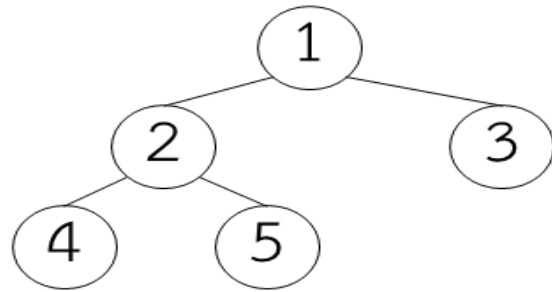
4 2 5 1 3

Postorder traversal

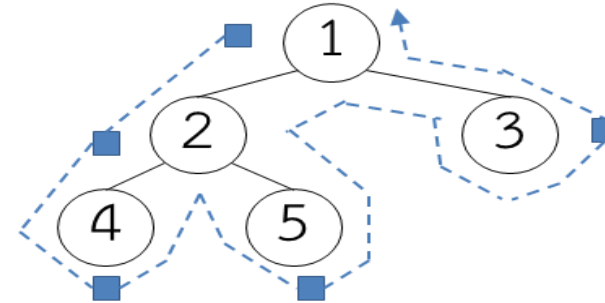
4 5 2 3 1

## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

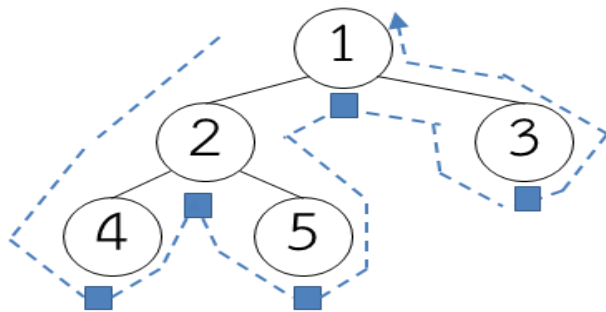
ผลลัพธ์การท่องเข้าไปในต้นไม้แบบทวิภาคแบบต่าง ๆ



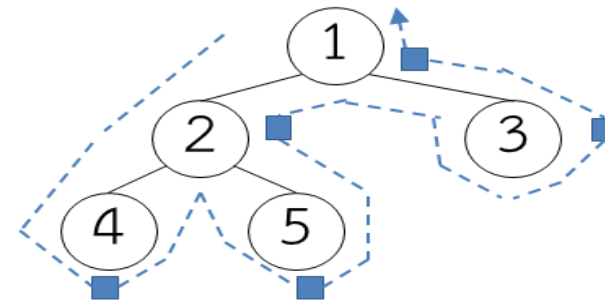
ลำดับข้อมูลต้นไม้เริ่มต้น



ลำดับข้อมูลการท่องเข้าไปในต้นไม้  
แบบ preorder



ลำดับข้อมูลการท่องเข้าไปในต้นไม้  
แบบ inorder



ลำดับข้อมูลการท่องเข้าไปในต้นไม้  
แบบ postorder



## 7.4 การท่องเที่ยวไปในต้นไม้แบบทวิภาค(ต่อ)

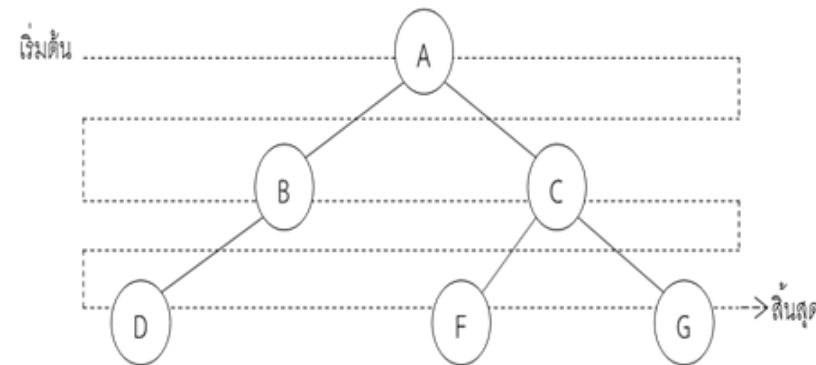
### 7.4.2 วิธีการท่องเที่ยวแบบแนวกว้าง

การดำเนินการในลักษณะท่องเที่ยวแบบลำดับชั้น คือ ท่องจากรูกุทโหนดและท่องลงมาที่ลำดับชั้นที่อยู่รองลงมาจากรูกุทโหนดและท่องจากโหนดด้านซ้ายมาโหนดด้านขวาตามลำดับ โดยมีลำดับการเข้าถึงแต่ละโหนดดังนี้

- 1) กำหนดโหนดเริ่มต้นก่อน นำเข้าคิวเพื่อทำเครื่องหมายว่าเข้ามาแล้ว
- 2) กำหนดโหนด adjacent ทั้งหมดของโหนดที่อยู่ในข้อ 1.
- 3) เดินจากซ้ายไปขวา (branch ซ้ายของ tree จะมีค่าน้อยกว่า branch ขวา)

## 7.4 การท่องเข้าไปในต้นไม้แบบทวิภาค(ต่อ)

ดังรูปภาพที่ 7.13



ภาพที่ 7.10 ผลลัพธ์ของการท่องแบบแนวกว้าง (breadth first traversals)

สามารถเข้าถึงข้อมูลแต่ละโหนด ได้ ดังนี้ A B C D F G

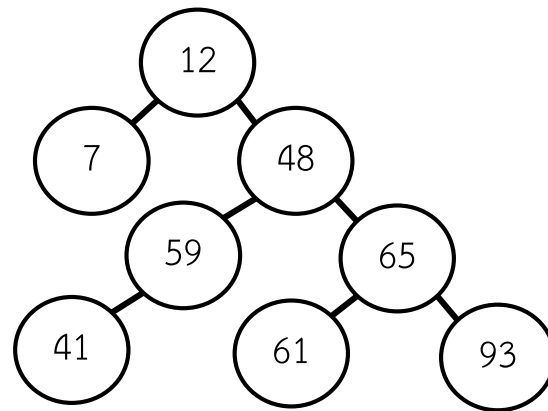


## 7.5 การค้นหาข้อมูลในต้นไม้แบบทวิภาค

- ต้นไม้แบบทวิภาคเป็นต้นไม้อีกรูปแบบหนึ่งที่มีรูปแบบในการพิจารณาเงื่อนไขในการเปรียบเทียบค่าของข้อมูลในแต่ละโหนดเพื่อช่วยให้การแยกแยะข้อมูลสามารถจัดการได้อย่างมีระเบียบและเพื่อช่วยให้เกิดความสะดวกในการค้นหาเพื่อนำไปใช้งาน
- โดยทั่วไปแล้วการค้นหาข้อมูลในแต่ละโหนดจะพบว่าเมื่อทำการค้นหาข้อมูลโหนดที่อยู่ด้านซ้ายจะมีค่าน้อยกว่าโหนดพ่อ และโหนดที่อยู่ด้านขวาจะมีค่ามากกว่าโหนดพ่อ

## 7.5 การค้นหาข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.5 การค้นหาค่า 93 ในต้นไม้แบบทวิภาค





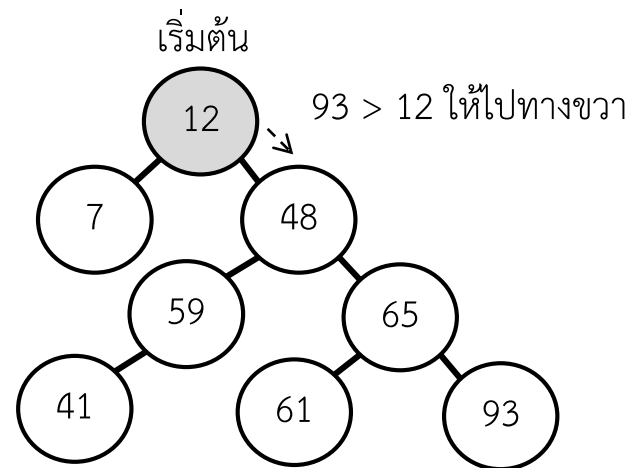
## 7.5 การค้นหาข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.5 การค้นหาค่า 93 ในต้นไม้แบบทวิภาค

ขั้นตอนการค้นหา

1) เริ่มต้นที่ root โหนด 12 เปรียบเทียบกับข้อมูลที่ต้องการค้นหา คือ  $93 > 12$  ซึ่งมีค่ามากกว่า

ให้ทำการค้นหาต่อไปทางด้านขวา

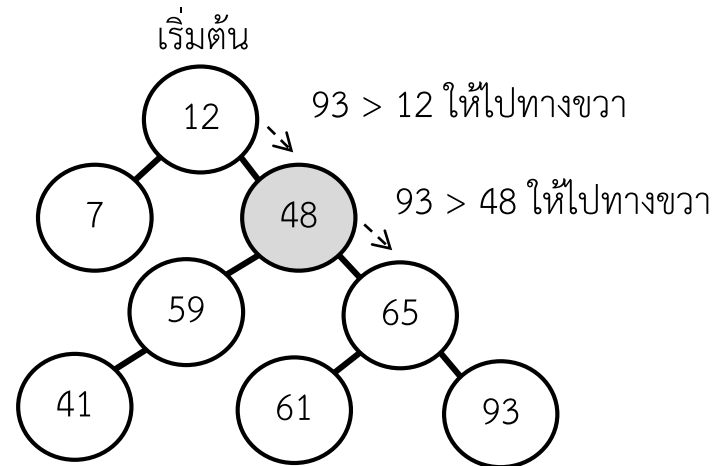


## 7.5 การค้นหาข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.5 การค้นหาค่า 93 ในต้นไม้แบบทวิภาค

ขั้นตอนการค้นหา

2) โหนด 48 เปรียบเทียบกับโหนด 93 ซึ่ง  $93 > 48$  ซึ่งมีค่ามากกว่า ให้ทำการค้นหาต่อไปทางด้านขวา

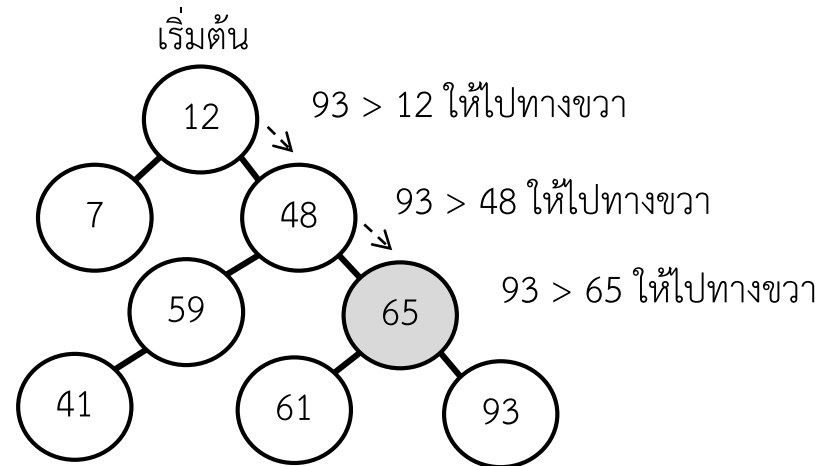


## 7.5 การค้นหาข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.5 การค้นหาค่า 93 ในต้นไม้แบบทวิภาค

ขั้นตอนการค้นหา

3) โหนด 65 เปรียบเทียบกับโหนด 93 ซึ่ง  $93 > 65$  ซึ่งมีค่ามากกว่า ให้ทำการค้นหาต่อไปทางด้านขวา

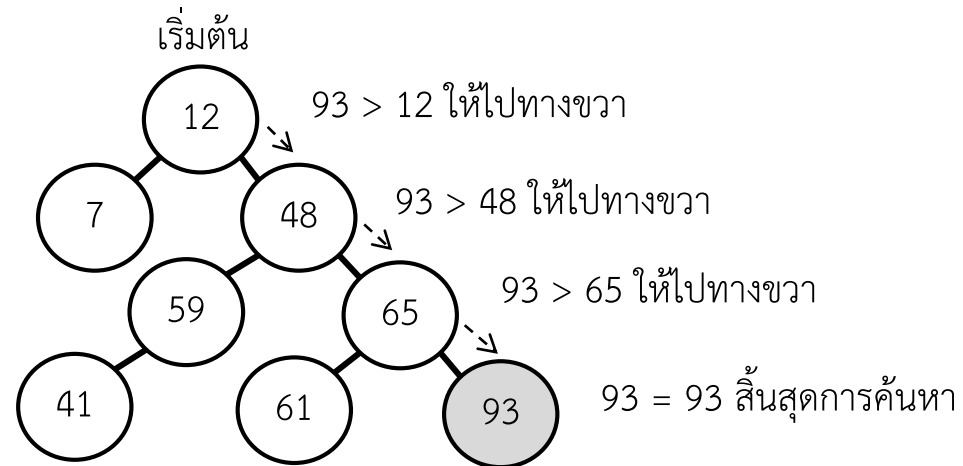


## 7.5 การค้นหาข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.5 การค้นหาค่า 93 ในต้นไม้แบบทวิภาค

ขั้นตอนการค้นหา

4) โหนด 93 เปรียบเทียบกับโหนดที่ต้องการค้นหา พบข้อมูลจึงสิ้นสุดการค้นหา





## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค

- การเพิ่มข้อมูลใหม่เข้าในต้นไม้แบบทวิภาคสิ่งที่ควรพิจารณาในการเพิ่มข้อมูลใหม่ คือ ตำแหน่งที่ต้องการนำข้อมูลนั้น ๆ เข้าในโครงสร้างต้นไม้ ซึ่งสามารถทำการพิจารณาอยู่ 2 กรณี คือ
  - กรณีการเพิ่มข้อมูลเข้าในต้นไม้ที่มีข้อมูลเดิมอยู่
  - กรณีการเพิ่มข้อมูลเข้าในต้นไม้ที่ยังไม่มีข้อมูลใด ๆ



## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

### ขั้นตอนการเพิ่มข้อมูลในต้นไม้แบบทวิภาค

1. ถ้าต้นไม้แบบทวิภาคยังไม่มีข้อมูล ให้โหนดที่เข้ามาใหม่เป็นรูทโหนดของต้นไม้แบบทวิภาค
2. ถ้าต้นไม้แบบทวิภาคมีข้อมูลอยู่ ให้พิจารณาโหนดที่เพิ่มเข้ามาใหม่ดังนี้
  - 2.1 ถ้าข้อมูลของโหนดที่เข้ามาใหม่ มากกว่า ข้อมูลเดิมของรูทโหนด ให้ทำการเพิ่มโหนดใหม่ลงในต้นไม้โหนดย่อยด้านขวา
  - 2.2 ถ้าข้อมูลของโหนดที่เข้ามาใหม่ น้อยกว่า ข้อมูลเดิมของรูทโหนด ให้ทำการเพิ่มโหนดใหม่ลงในต้นไม้โหนดย่อยด้านซ้าย



## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค(ต่อ)


ตัวอย่างที่ 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค

โดยใช้ข้อมูลตัวเลขจำนวน 10 จำนวนดังนี้

12 48 7 65 59 61 41 93 74 63

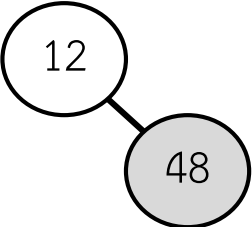
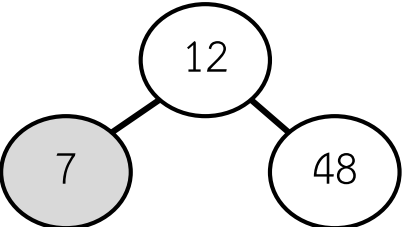
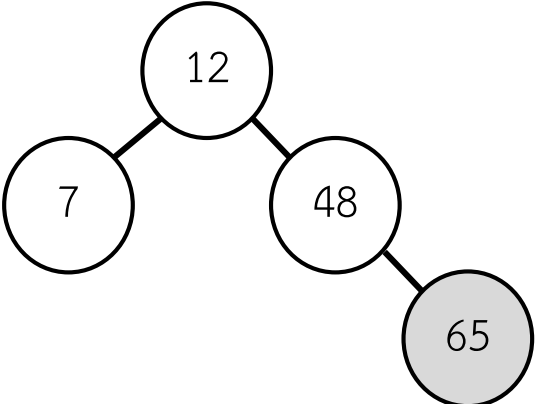
ขั้นตอนการเพิ่มข้อมูลต้นไม้แบบทวิภาค มีขั้นตอนการทำงานดังนี้

กำหนดให้โหนดแรกเป็นรูทโหนด

ลำดับ	รูปภาพ	การดำเนินการ
1.		เริ่มต้นการสร้างโหนดแรกในทรี ข้อมูลแรก คือ 12 และกำหนดให้เป็นรูทโหนด

## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

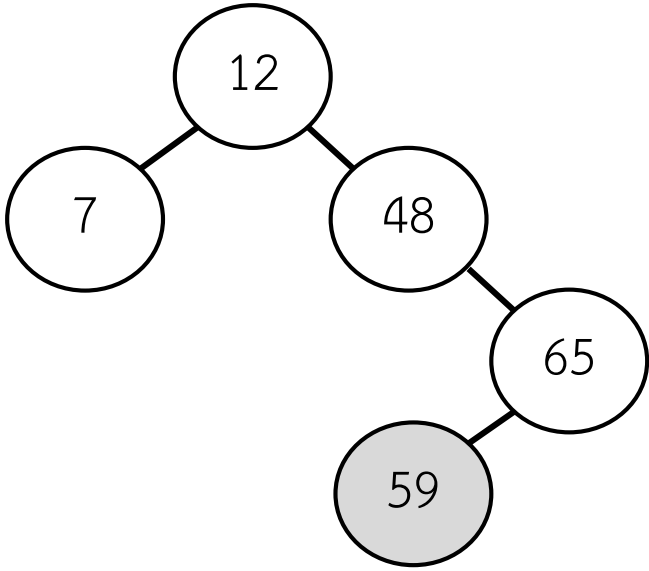
ตัวอย่างที่ 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค (ต่อ)

ลำดับ	รูปภาพ	การดำเนินการ
2.		<p>เพิ่มโหนด 48</p> <p>เมื่อเปรียบเทียบโหนด 48</p> <p>กับรูทโหนดมีค่ามากกว่า 12</p> <p>ดังนั้นแทรกโหนดทางด้านขวาของ 12</p>
3.		<p>เพิ่มโหนด 7</p> <p>เมื่อเปรียบเทียบโหนด 7</p> <p>กับรูทโหนดมีค่าน้อยกว่า 12</p> <p>ดังนั้นแทรกโหนดทางด้านซ้ายของ 12</p>
4.		<p>เพิ่มโหนด 65</p> <p>เมื่อเปรียบเทียบโหนด 65</p> <p>กับรูทโหนดมีค่ามากกว่า 12 และ 48</p> <p>ดังนั้นแทรกโหนดทางด้านขวาของ 48</p>



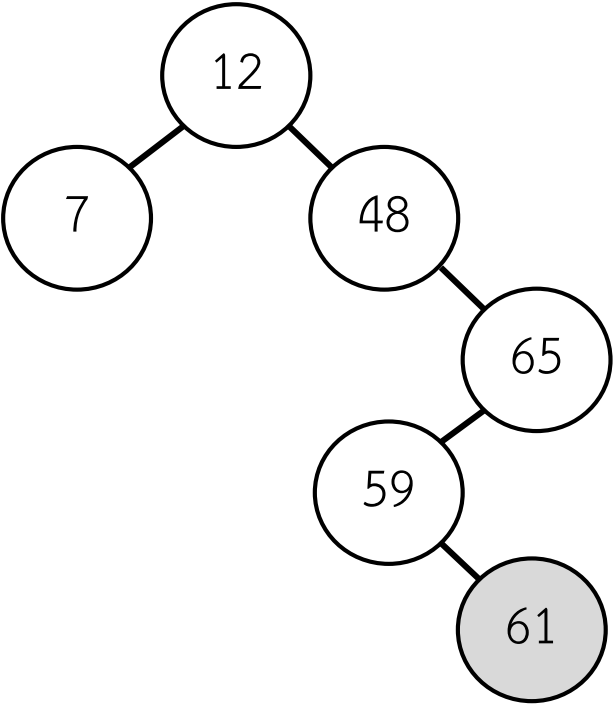
## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค (ต่อ)

ลำดับ	รูปภาพ	การดำเนินการ
5.		<p>เพิ่มโหนด 59</p> <p>เมื่อเปรียบเทียบโหนด 59</p> <p>กับรูทโหนดมีค่ามากกว่า 12, 48 และ 65</p> <p>ดังนั้นแทรกโหนดทางด้านซ้ายของ 65</p>

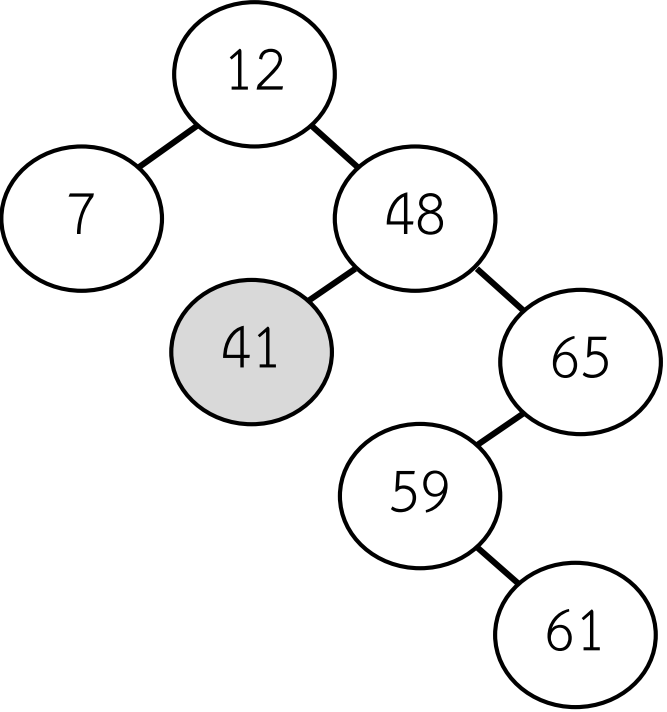
## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค (ต่อ)

ลำดับ	รูปภาพ	การดำเนินการ
6.		<p>เพิ่มโหนด 61          เมื่อเปรียบเทียบโหนด 61          กับรูทโหนดมีค่ามากกว่า 12, 48          แต่น้อยกว่า 65 และมากกว่า 59          ดังนั้นแทรกโหนดทางด้านขวาของ 59</p>

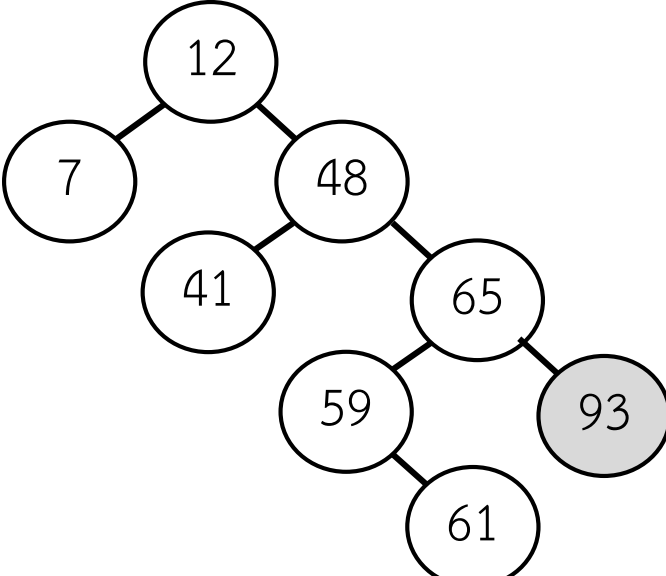
## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค (ต่อ)

ลำดับ	รูปภาพ	การดำเนินการ
7.		<p>เพิ่มโหนด 41          เมื่อเปรียบเทียบโหนด 41          กับรูทโหนดมีค่ามากกว่า 12          แต่น้อยกว่า 48          ดังนั้นแทรกโหนดทางด้านซ้ายของ 48</p>

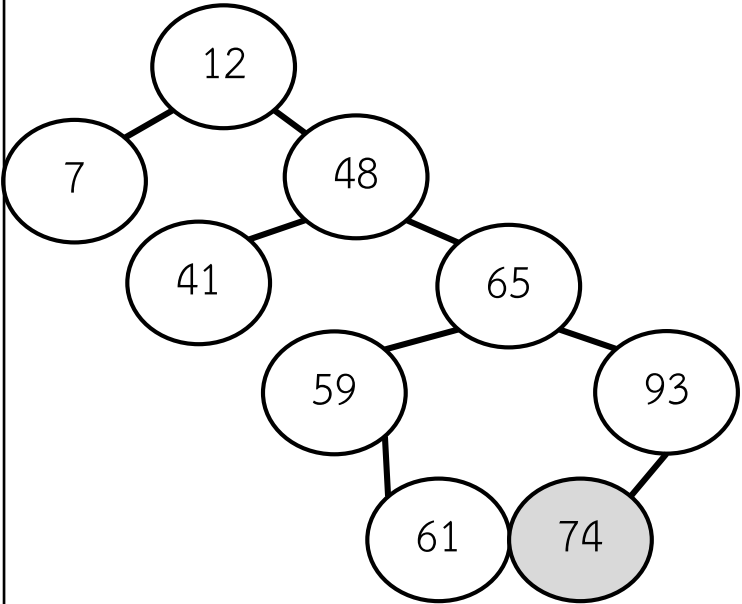
## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค (ต่อ)

ลำดับ	รูปภาพ	การดำเนินการ
8.	 <pre> graph TD     7((7)) --- 12((12))     12 --- 48((48))     48 --- 41((41))     48 --- 65((65))     65 --- 59((59))     65 --- 93((93))     59 --- 61((61))     style 93 fill:#ccc         </pre>	<p>เพิ่มโหนด 93          เมื่อเปรียบเทียบโหนด 93          กับรูทโหนดมีค่ามากกว่า 12, 48          แต่มากกว่า 65          ดังนั้นแทรกโหนดทางด้านขวาของ 65</p>

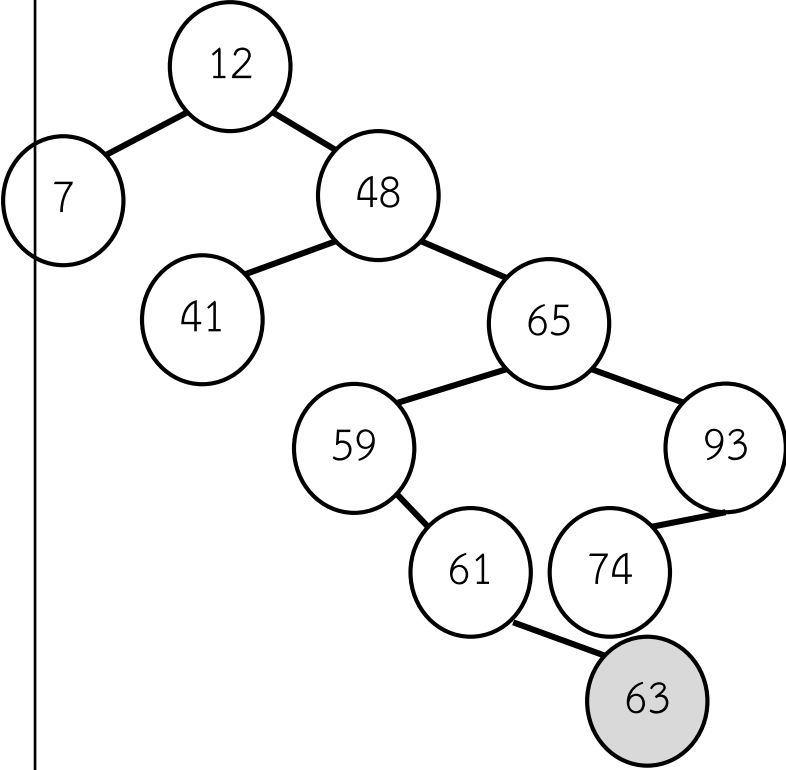
## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค (ต่อ)

ลำดับ	รูปภาพ	การดำเนินการ
9.	 <pre> graph TD     7((7)) --- 12((12))     7 --- 41((41))     12 --- 48((48))     48 --- 65((65))     41 --- 59((59))     59 --- 61((61))     65 --- 93((93))     93 --- 74((74))             </pre>	<p>เพิ่มโหนด 74                      เมื่อเปรียบเทียบโหนด 74                      กับรูทโหนดมีค่ามากกว่า 12, 48 และ 65                      แต่น้อยกว่า 93                      ดังนั้นแทรกโหนดทางด้านซ้ายของ 93</p>

## 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.6 การเพิ่มข้อมูลในต้นไม้แบบทวิภาค (ต่อ)

ลำดับ	รูปภาพ	การดำเนินการ
10.	 <pre> graph TD     7((7)) --- 12((12))     12 --- 48((48))     48 --- 41((41))     48 --- 65((65))     65 --- 59((59))     65 --- 93((93))     59 --- 61((61))     93 --- 74((74))     61 --- 63((63))     style 63 fill:#ccc             </pre>	<p>เพิ่มโหนด 63                  เมื่อเปรียบเทียบโหนด 63                  กับรูทโหนดมีค่ามากกว่า 12, 48                  แต่น้อยกว่า 65 และมากกว่า 61                  ดังนั้นแทรกโหนดทางด้านขวาของ 61</p>



## 7.7 การลบน้ำหนคจากต้นไม้แบบทวิภาค

- การลบน้ำหนคจากต้นไม้ค่อนข้างมีล้าดับชั้นตอนทีซ้บซ้อน เพราะชั้นตอนการเชื่อมโยงทำได้ยากกว่า ในการลบน้ำหนคต่าง ๆ ต้องพิจารณา โหนคเดิมทีมีล้าดับถ้ดจากโหนคทีลบ

ดังนั้น การลบน้ำหนคจากต้นไม้จึงต้องแยกพิจารณาเป็นประเด็นหลัก ๆ ดังนี้

- กรณีลบน้ำหนคทีปลายโหนค
- กรณีลบน้ำหนคทีมีลูกเพียงหนึ่งโหนค



## 7.7 การลบโหนดจากต้นไม้แบบทวิภาค(ต่อ)

### 7.7.1 กรณีลบโหนดที่ปลายโหนด

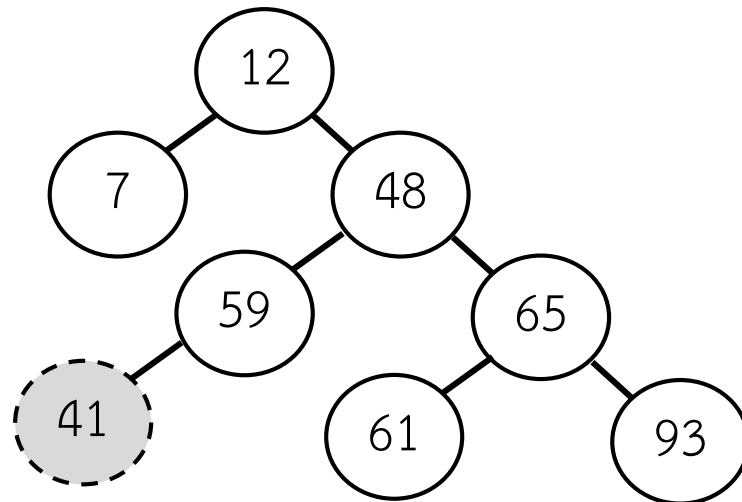
เป็นการลบข้อมูลที่ง่ายที่สุด เนื่องจากการลบโหนดใบที่อยู่ปลายสุดของต้นไม้ และให้พอยน์เตอร์ที่ชี้ไปที่โหนดที่จะลบชี้ไปที่ NULL แทน ตัวอย่างการลบปลายโหนด (โหนด 41)



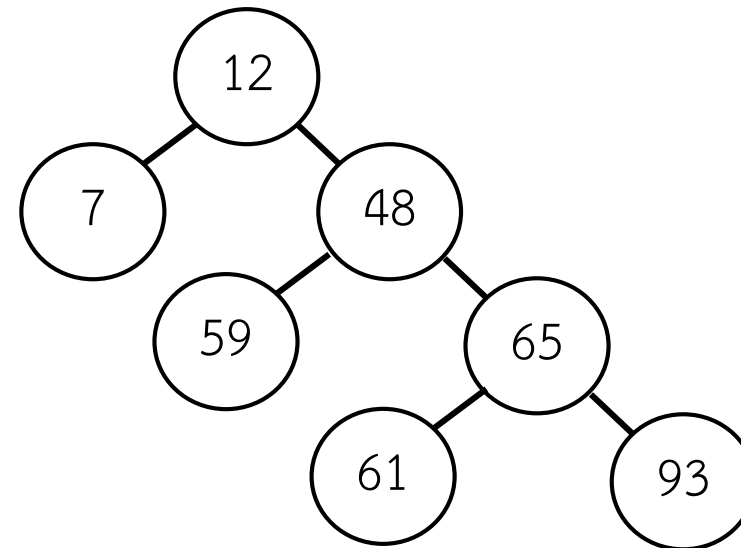
## 7.7 การลบโหนดจากต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.7 การลบปลายโหนด (โหนด 41)

กำหนดให้โหนดแม่ที่อ้างอิงโหนดใบที่เป็นโหนดลูกและโหนดที่ต้องการลบมีค่าเท่ากับ null ยกเลิกการอ้างอิงไปยังโหนดใบ



ก่อนการลบโหนด 41



หลังการลบโหนด 41



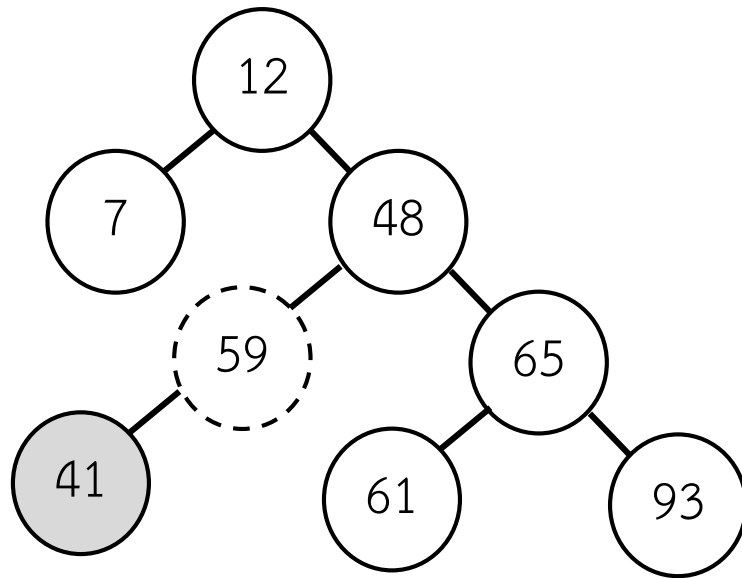
## 7.7 การลบโหนดจากต้นไม้แบบทวิภาค(ต่อ)

### 7.7.2 กรณีลบโหนดที่มีลูกเพียงหนึ่งโหนด

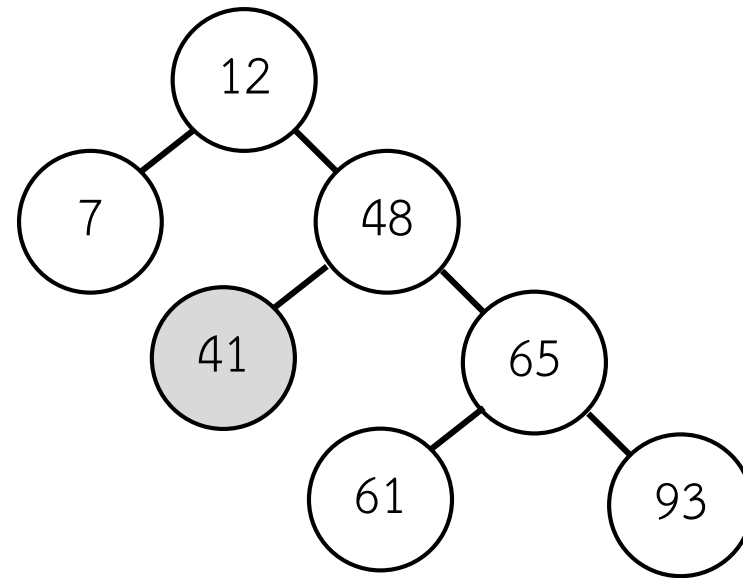
เป็นการลบข้อมูลของโหนดที่ยังมีโหนดลูกด้านซ้ายหรือด้านขวาเพียงหนึ่งตัวเท่านั้น การลบโหนดนั้นจะทำการลบโหนดที่ต้องการและให้พอยน์เตอร์ทำการชี้ไปยังโหนดลูกนั้น ๆ แทนขึ้นมาแทนโหนดที่ทำการลบไปนั่นเอง ดังตัวอย่างแสดงการลบโหนด 59 ของต้นไม้

## 7.7 การลบโหนดจากต้นไม้แบบทวิภาค(ต่อ)

ตัวอย่างที่ 7.8 ต้องการลบโหนด 59 สามารถทำได้โดยให้โหนด 48 ชี้ไปที่โหนด 41 ได้เลย



ก่อนการลบโหนด 59



หลังการลบโหนด 59



มหาวิทยาลัยราชภัฏนครปฐม