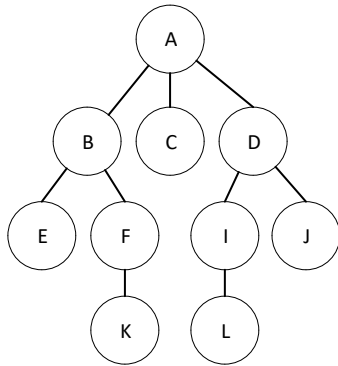


Assignment: TREE

1. อธิบายขั้นตอนโดยละเอียดในการแปลง General tree ให้เป็น Binary tree โดยใช้วิธีการแทนแบบ left: first child right: next sibling แสดงภาพประกอบในแต่ละขั้นตอนของกระบวนการแปลง

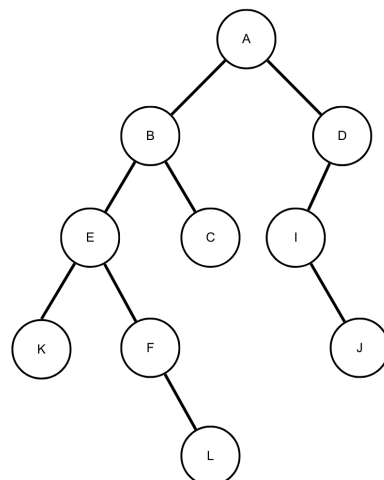


จากภาพ General Tree ที่มีโหนด A, B, C, D, E, F, I, J, K, L

1. ให้เริ่มจากโหนดราก (A) และเชื่อมต่อโหนดลูกคนแรก (B) เป็นลูกทางซ้าย
2. เชื่อมโหนดถัดไปของ B (ซึ่งคือ C) เป็นลูกทางขวาของ B
3. ทำเช่นนี้ต่อไปเรื่อยๆ สำหรับโหนดอื่นๆ: C เชื่อมกับ D ทางขวา
4. จากนั้นไปที่ B จะมีลูกคนแรกคือ E ให้เชื่อม E เป็นลูกทางซ้ายของ B
5. F เป็นพี่น้องของ E ให้เชื่อม F ทางขวาของ E
6. ไปที่ D มีลูกคนแรกคือ I และพี่น้องคือ J ให้เชื่อมทางซ้ายและขวาตามลำดับ
7. E มีลูกคือ K ให้เชื่อม K เป็นลูกทางซ้าย
8. F มีลูกคือ L ให้เชื่อม L เป็นลูกทางขวาของ F

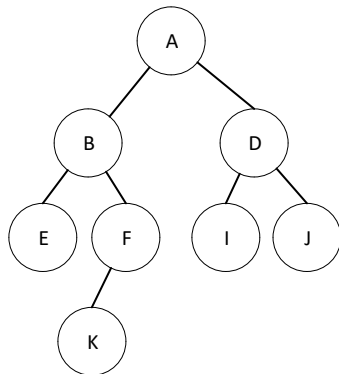
Binary Tree ที่ได้จะมีการเชื่อมต่อแบบนี้

- A -> B (left)
- B -> C (right)
- B -> E (left)
- E -> K (left)
- E -> F (right)
- D -> I (left)
- I -> J (right)
- F -> L (right)



Binary Tree

2. จาก tree ดังกล่าวให้ แสดงการเดินทางใน tree ด้วยวิธีการ



2.1. Pre-order

ขั้นตอนการเดินทาง Pre-order (NLR: Node-Left-Right)

1. เริ่มจาก **Root: A**
2. ไปทางซ้ายของ **A: B**
3. ไปทางซ้ายของ **B: E**
4. กลับไปที่ **B** และไปทางขวา: **F**
 - ไปทางซ้ายของ **F: K**
5. กลับไปที่ **A** และไปทางขวา: **D**
6. ไปทางซ้ายของ **D: I**
7. กลับไปที่ **D** และไปทางขวา: **J**

ผลลัพธ์ Pre-order: A, B, E, F, K, D, I, J

2.2.In-order

ขั้นตอนการเดินทาง In-order (LNR: Left-Node-Right)

1. เริ่มจาก **Root: A**
2. ไปทางซ้ายของ **A: B**
3. ไปทางซ้ายของ **B: E**
4. กลับไปที่ **B**
5. ไปทางขวาของ **B: F**
 - ไปทางซ้ายของ **F: K**
 - กลับไปที่ **F**
6. กลับไปที่ **A**
7. ไปทางขวาของ **A: D**
8. ไปทางซ้ายของ **D: I**
9. กลับไปที่ **D**
10. ไปทางขวาของ **D: J**

ผลลัพธ์ In-order: E, B, K, F, A, I, D, J

2.3.Post-order

ขั้นตอนการเดินทาง **Post-order (LRN: Left-Right-Node)**

1. เริ่มจาก **Root: A**
2. ไปทางซ้ายของ **A: B**
3. ไปทางซ้ายของ **B: E**
4. กลับไปที่ **B** และไปทางขวา: **F**
 - ไปทางซ้ายของ **F: K**
 - กลับไปที่ **F** แล้วจึง **F**
5. กลับไปที่ **B** แล้วจึง **B**
6. กลับไปที่ **A** และไปทางขวา: **D**
7. ไปทางซ้ายของ **D: I**
8. กลับไปที่ **D** และไปทางขวา: **J**
9. กลับไปที่ **D** แล้วจึง **D**
10. กลับไปที่ **A** แล้วจึง **A**

ผลลัพธ์ Post-order คือ: E, K, F, B, I, J, D, A

3. จากผลการเดินทางแบบ
in-order: [D, B, E, A, F, C] และ
pre-order [A, B, D, E, C, F]

ให้แสดงวิธีการเป็นขั้นเป็นตอนในการสร้าง binary tree จากผลการเดินทางดังกล่าว

ผลการเดินทาง

- In-order: D, B, E, A, F, C
- Pre-order: A, B, D, E, C, F

วิธีการสร้าง

1. เริ่มที่ Pre-order ลำดับแรกคือ A เป็น root
2. หาดำแหน่ง A ใน In-order (อยู่ลำดับที่ 4) แบ่งเป็นซ้าย (D, B, E) และขวา (F, C)
3. ดู Pre-order ลำดับถัดไปคือ B ใส่เป็นลูกซ้ายของ A
4. B ใน In-order อยู่ลำดับที่ 2 แบ่งซ้าย (D) ขวา (E)
5. ใส่ D เป็นลูกซ้ายของ B และ E เป็นลูกขวาของ B
6. กลับมาที่ฝั่งขวาของ A ใส่ F เป็นลูกซ้ายของ C ตาม Pre-order

4. จากผลการเดินทางแบบ
in-order: [4, 2, 5, 1, 6, 3, 7] และ
post-order: [4, 5, 2, 6, 7, 3, 1]

ให้แสดงวิธีการเป็นขั้นเป็นตอนในการสร้าง binary tree จากผลการเดินทางดังกล่าว

ผลการเดินทาง:

- In-order: 4, 2, 5, 1, 6, 3, 7
- Post-order: 4, 5, 2, 6, 7, 3, 1

วิธีการสร้าง:

1. เริ่มที่ Post-order ลำดับสุดท้ายคือ 1 เป็น root
2. หาดำแหน่ง 1 ใน In-order (อยู่ลำดับที่ 4) แบ่งเป็นซ้าย (4, 2, 5) และขวา (6, 3, 7)
3. ดู Post-order ลำดับถัดไปคือ 3 ใส่เป็นลูกขวาของ 1
4. 3 ใน In-order อยู่ลำดับที่ 6 แบ่งซ้าย (6) ขวา (7)
5. ใส่ 6 เป็นลูกซ้ายของ 3 และ 7 เป็นลูกขวาของ 3
6. ฝั่งซ้ายของ 1: ดู Post-order ใส่ 2 เป็นลูกซ้ายของ 1
7. 2 ใน In-order อยู่ลำดับที่ 2 ใส่ 4 เป็นลูกซ้ายของ 2 และ 5 เป็นลูกขวา