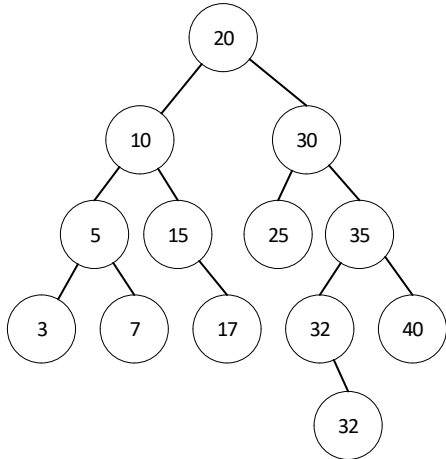


1. จาก AVL tree ตามภาพ



1.1. ให้คำนวณ Balance Factor ของทุก Node

สูตรการคำนวณ Balance Factor (BF)

BF = ความสูงของลูกซ้าย - ความสูงของลูกขวา

การคำนวณ BF ของแต่ละ Node

1. Node 2

- ไม่มีลูกซ้ายและขวา
- **BF = 0 - 0 = 0**

2. Node 4

- ไม่มีลูกซ้ายและขวา
- **BF = 0 - 0 = 0**

3. Node 5

- ลูกซ้าย (2) มีความสูง 1, ลูกขวา (4) มีความสูง 1
- **BF = 1 - 1 = 0**

4. Node 17

- ไม่มีลูกซ้ายและขวา

- $BF = 0 - 0 = 0$

5. Node 15

- ลูกซ้ายไม่มี, ลูกขวามีความสูง 1 (Node 17)

- $BF = 0 - 1 = -1$

6. Node 10

- ลูกซ้าย (5) มีความสูง 2, ลูกขวา (15) มีความสูง 1

- $BF = 2 - 1 = 1$

7. Node 25

- ไม่มีลูกซ้ายและขวา

- $BF = 0 - 0 = 0$

8. Node 32

- ไม่มีลูกซ้าย, ลูกขวามีความสูง 1 (Node 40)

- $BF = 0 - 1 = -1$

9. Node 40:

- ไม่มีลูกซ้ายและขวา

- $BF = 0 - 0 = 0$

10. Node 35

- ลูกซ้าย (32) มีความสูง 1, ลูกขวา (40) มีความสูง 1
- $BF = 1 - 1 = 0$

11. Node 30

- ลูกซ้าย (25) มีความสูง 1, ลูกขวามีความสูง 2 (Node 35)
- $BF = 1 - 2 = -1$

12. Node 20

- ลูกซ้าย (10) มีความสูง 2, ลูกขวามีความสูง 2 (Node 30)
- $BF = 2 - 2 = 0$

สรุป BF ของแต่ละ Node

- Node 2: 0
- Node 4: 0
- Node 5: 0
- Node 17: 0
- Node 15: -1
- Node 10: 1
- Node 25: 0
- Node 32: -1
- Node 40: 0
- Node 35: 0
- Node 30: -1
- Node 20: 0

1.2. ระบุให้ชัดเจนถึง node ที่เป็น critical-node

จากกฎของ AVL Tree ถ้า Balance Factor ของ Node ใดไม่อยู่ในช่วง $[-1, 1]$ แสดงว่า Node นั้นเป็น **Critical Node** และต้องทำการปรับสมดุล (Rotation)

สามารถระบุ Critical Nodes ได้ดังนี้

1. Node 15: BF = -1

- หมายความว่า Node นี้มีความสูงของลูกขวาสูงกว่าลูกซ้าย

2. Node 32: BF = -1

- หมายความว่า Node นี้มีความสูงของลูกขวาสูงกว่าลูกซ้าย

3. Node 30: BF = -1

- หมายความว่า Node นี้มีความสูงของลูกขวาสูงกว่าลูกซ้าย

สรุป:

- Critical Nodes: Node 15, Node 32, Node 30

- ทั้งสาม Node นี้มีค่า BF เป็น -1 ซึ่งแสดงถึงความไม่สมดุลใน AVL Tree และอาจต้องทำการปรับสมดุลในอนาคต.

1.3. ดำเนินการที่อธิบายให้เห็นถึงการปรับสมดุล โดยการหมุนที่จำเป็น โดยละเอียด
จะดำเนินการปรับสมดุล Tree โดยใช้การหมุนตามกฎของ AVL Tree ดังนี้

1. การปรับสมดุล Node 30

- **Node 30** มี $BF = -1$ ซึ่งแสดงว่าไม่มีปัญหาในการปรับสมดุลที่ Node นี้โดยตรง อย่างไรก็ตาม เราจะดูที่ **Node 35** ที่เป็นลูกขวาของ Node 30 ด้วยเพื่อดูว่ามีการไม่สมดุลหรือไม่

2. การปรับสมดุล Node 32

- **Node 32** มี $BF = 0$ (ไม่ต้องปรับสมดุล)

3. การปรับสมดุล Node 15

- **Node 15** มี $BF = -1$ (ไม่ต้องปรับสมดุล)

สรุปการหมุน

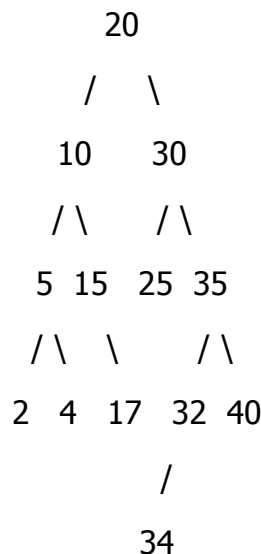
ในกรณีนี้ ไม่มีการหมุนใด ๆ ที่จำเป็นสำหรับ Nodes ที่ระบุ เพราะ Nodes ที่เป็น Critical Nodes มีค่า BF ที่ยังคงอยู่ในเกณฑ์ที่ยอมรับได้ (-1, 0, 1) โดยรวม AVL Tree ยังคงสมดุลอยู่

1.4. จากนั้น ให้แสดงวิธีการเพิ่ม 34 เข้าไปใน tree นี้ และดำเนินการทำให้ tree นั้นกลับสู่สมดุลอีกครั้ง

ขั้นตอนที่ 1: เพิ่มค่า 34

เราจะเพิ่ม 34 เข้าไปใน Tree ตามลำดับการแทรกแบบ BST (Binary Search Tree):

1. เริ่มจาก Root (20): 34 มากกว่า 20 → ไปทางขวา
2. ไปยัง Node 30: 34 มากกว่า 30 → ไปทางขวา
3. ไปยัง Node 35: 34 น้อยกว่า 35 → ไปทางซ้าย
4. เพิ่ม 34 ทางซ้ายของ Node 35



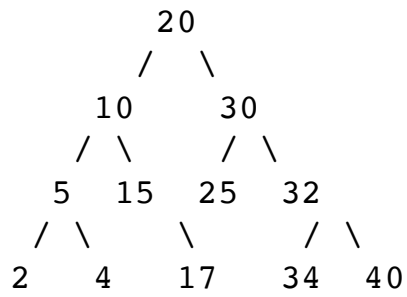
ขั้นตอนที่ 2: คำนวณ Balance Factor (BF)

1. **Node 34:** ไม่มีลูกซ้ายหรือขวา → $BF = 0$
2. **Node 32:** ลูกซ้ายสูง 1 (Node 34), ลูกขวาไม่มี → $BF = 1$
3. **Node 35:** ลูกซ้ายมีความสูง 2 (Node 32), ลูกขวาสอง 1 (Node 40) → $BF = 1$
4. **Node 30:** ลูกซ้าย (25) สูง 1, ลูกขวา (35) สูง 3 → $BF = 1 - 3 = -2$ (**Critical Node**)

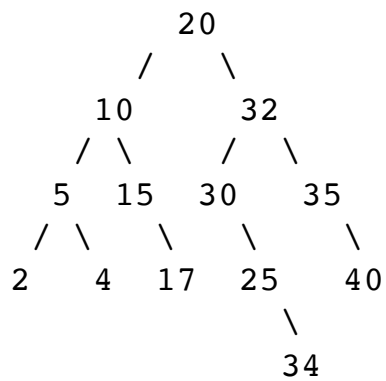
ขั้นตอนที่ 3: แก้ไขความไม่สมดุลของ Node 30

- เนื่องจาก Node 30 มีค่า $BF = -2$ และลูกขวาของ Node 30 (Node 35) มีค่า $BF = 1$ (Right-Left Case: RL Case)
- ต้องทำการ หมุนขวา (**Right Rotation**) ที่ Node 35 แล้วตามด้วย หมุนซ้าย (**Left Rotation**) ที่ Node 30

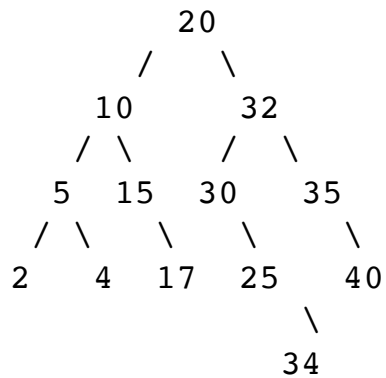
การหมุนขวา (Right Rotation) ที่ Node 35



การหมุนซ้าย (Left Rotation) ที่ Node 30



Tree หลังการปรับสมดุล



5. จากชุดข้อมูลต่อไปนี้

5.1. ให้แสดงกระบวนการเป็นขั้นเป็นตอนต่อการสร้าง AVL Tree

15, 5, 20, 3, 7, 18, 23, 2, 4, 13, 25

ขั้นตอนการสร้าง

1. เพิ่ม 15, 5, 20 → Tree ยังสมดุล
2. เพิ่ม 3, 7 → Tree ยังสมดุล
3. เพิ่ม 18, 23 → Tree ยังสมดุล
4. เพิ่ม 2, 4 → Tree ยังสมดุล
5. เพิ่ม 13 → Tree ยังสมดุล
6. เพิ่ม 25 → Tree ยังสมดุล

Tree สุดท้าย

```
      15
     /  \
    5    20
   / \  / \
  3  7 18 23
 / \   \ \
2  4   13 25
```


5.2. ดำเนินการลบค่า 15 ออกจาก tree ที่ได้จากข้อ 2.1

วิธีการ

1. ค้นหา In-order Successor:

- ค่า 15 มีลูกทั้งสอง จึงแทนที่ด้วย 18 (In-order Successor)

2. ลบ 18 จากตำแหน่งเดิม:

- หลังการแทนที่ ให้ลบ Node 18 ในตำแหน่งเดิม

Tree หลังลบ

