

AVL Tree \Leftrightarrow ① BST

and ② 각 노드의

balance factor

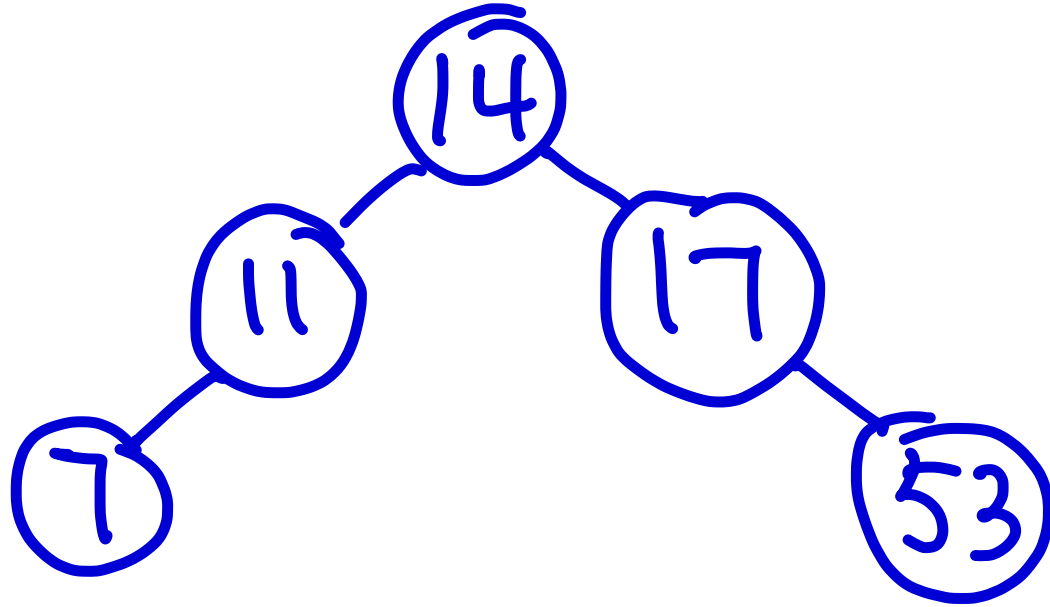
$\in \{-1, 0, 1\}$

and ③ balance factor

= height of left subtree

- height of right subtree

삽입시 LL Rot2



여기에 4 삽입:

1) 삽입된 노드부터
부모로 올라가며
balance factor 확인

2) BF가 깨졌으면
왜 깨졌나?

3) 회전

① Rotation 대상

서 노드의 중앙값을 루트로

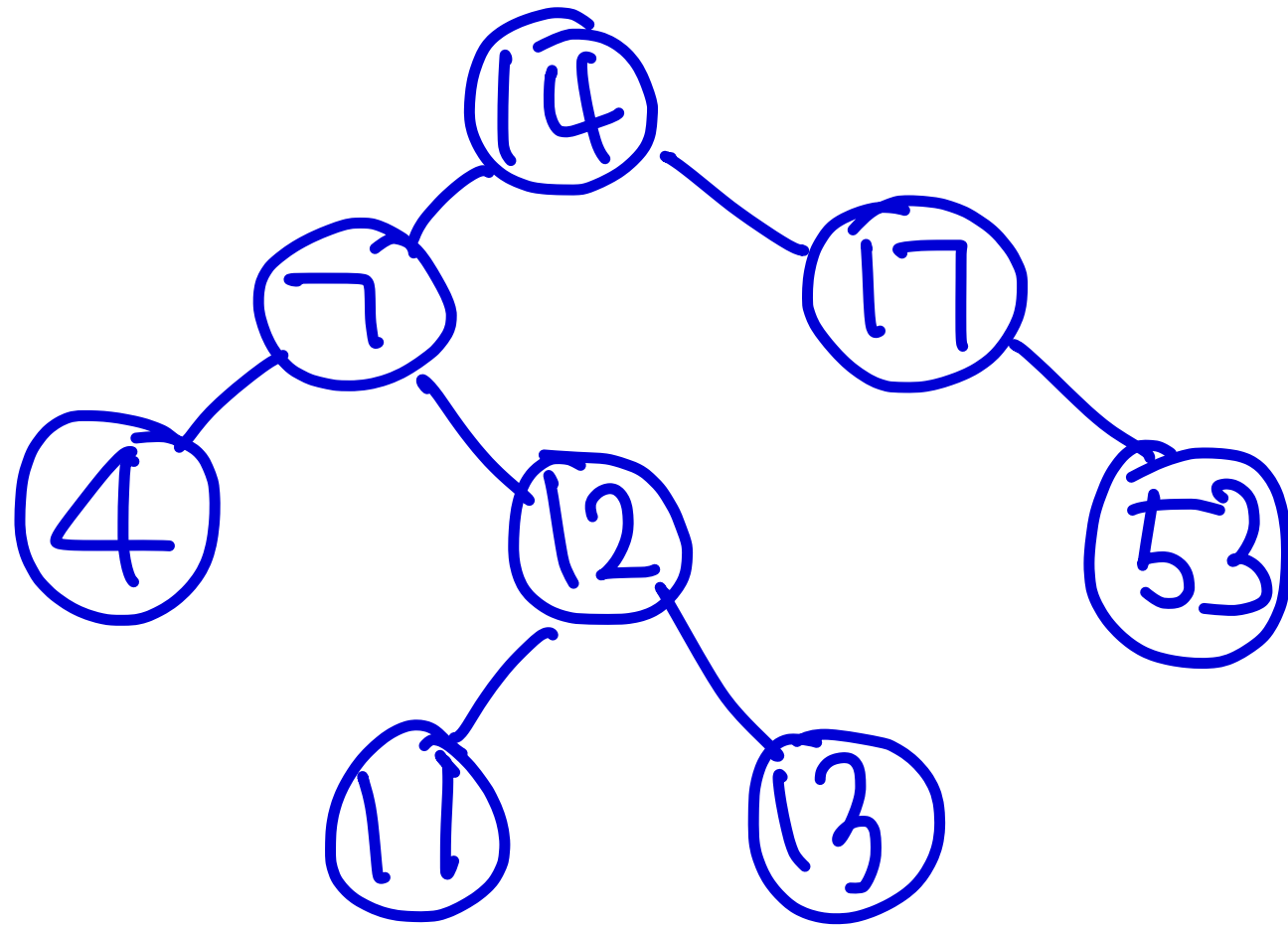
② 나머지 노드를 재배치

4) 회전 완료 후

전체 bf 다시 확인

AVL tree인가?

RL Rotate



여기에 8 삽입

1) 삽입된 노드부터
부모로 올라가며
balance factor 확인

2) BF가 깨졌으면
왜 깨졌나?

3) 회전

① Rotation 대상

서 노드의 중앙값을 루트로

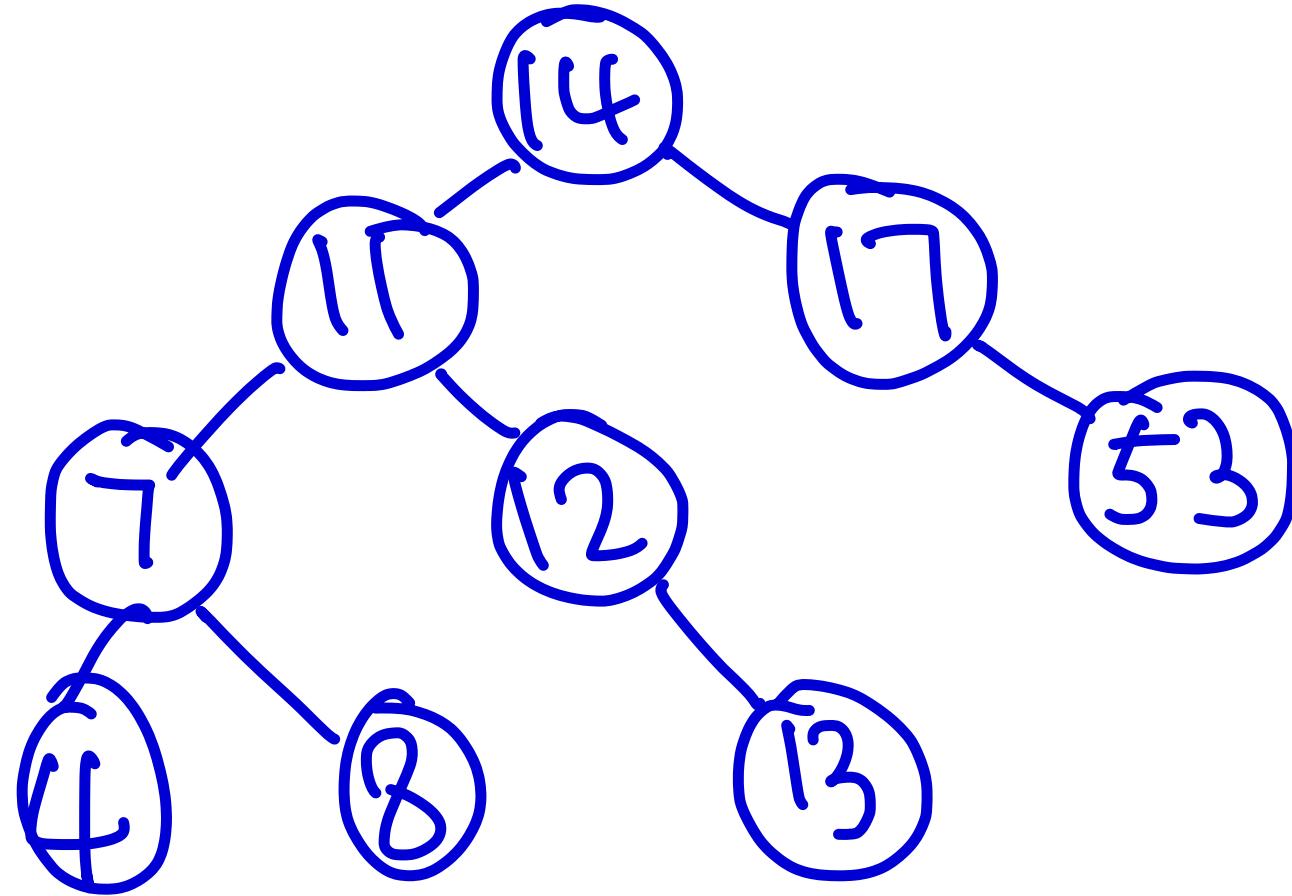
② 나머지 노드를 재배치

4) 회전 완료 후

전체 bf 다시 확인

AVL tree인가?

RR Rota



여기에 60추가

1) 삽입된 노드부터
부모로 올라가며
balance factor 확인

2) BF가 깨졌으면
대개졌나?

3) 회전

① Rotation 대상

서 노드의 중앙값을 루트로

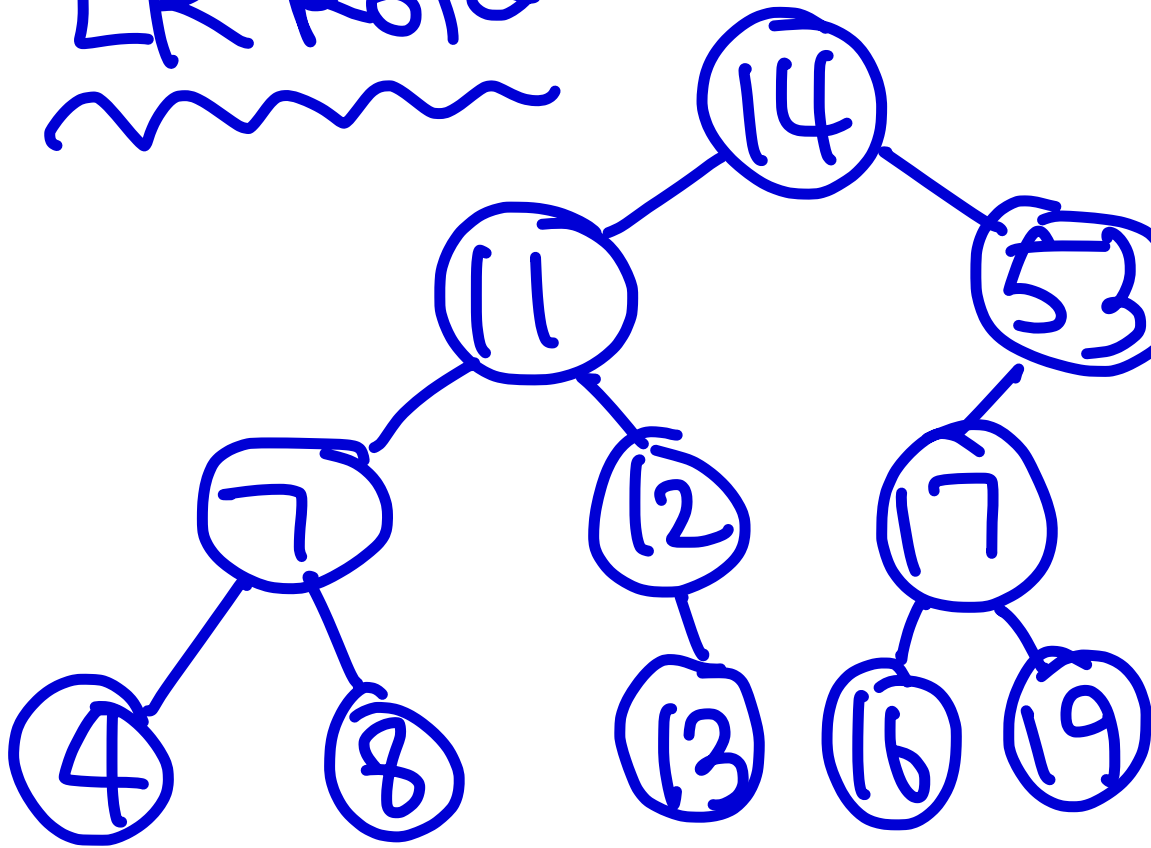
② 나머지 노드를 재배치

4) 회전 완료 후

전체 bf 다시 확인

AVL tree인가?

LR Rot2



여기까지 20 삽입

1) 삽입된 노드부터
부모로 올라가며
balance factor 확인

2) BF가 깨졌으면
왜 깨졌나?

3) 회전

① Rotation 대상

서 노드의 중앙값을 루트로

② 나머지 노드를 재배치

4) 회전 완료 후

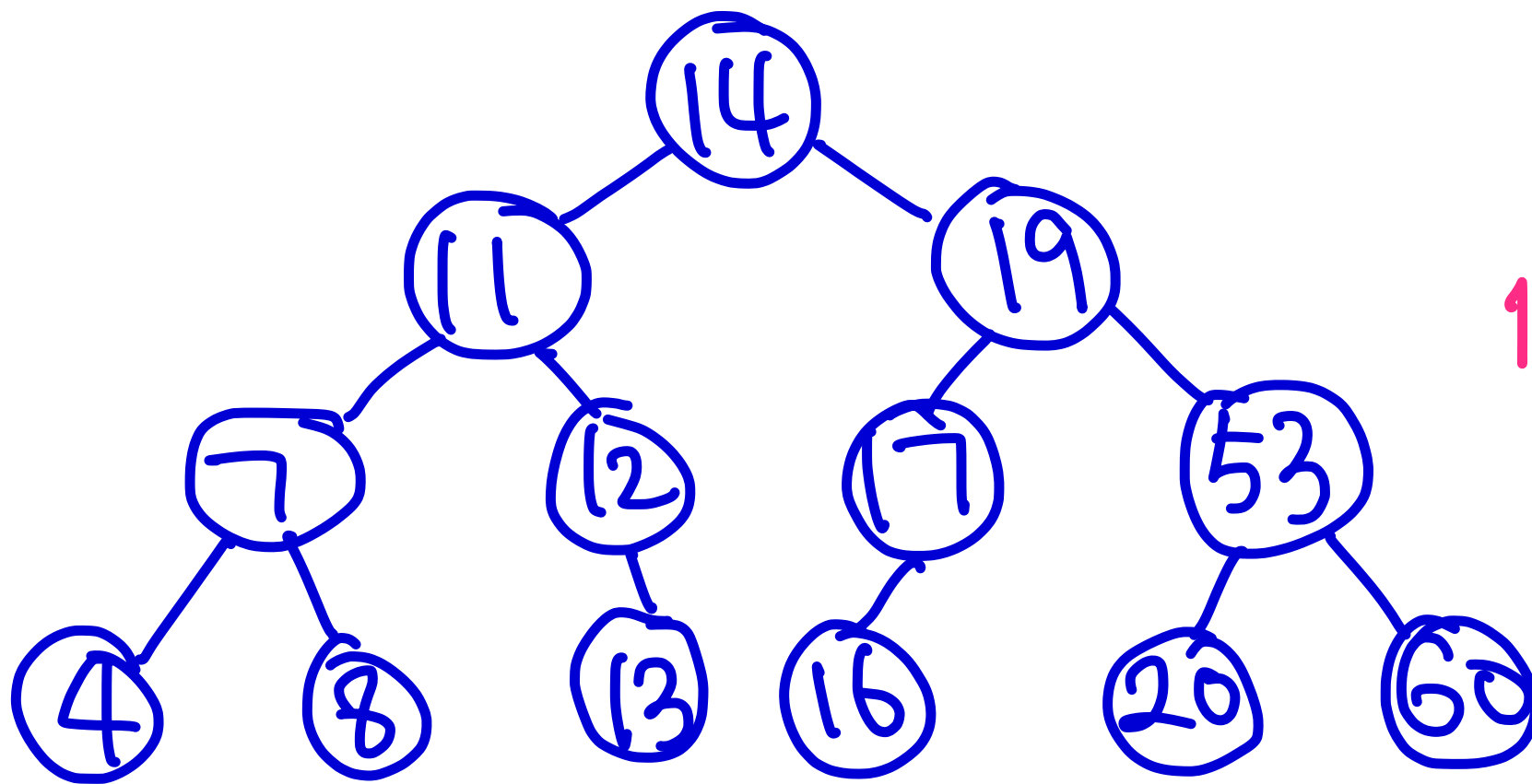
전체 bf 다시 확인

AVL tree인가?

제거

AVL Tree의 제거 \Leftrightarrow BST에서의 제거

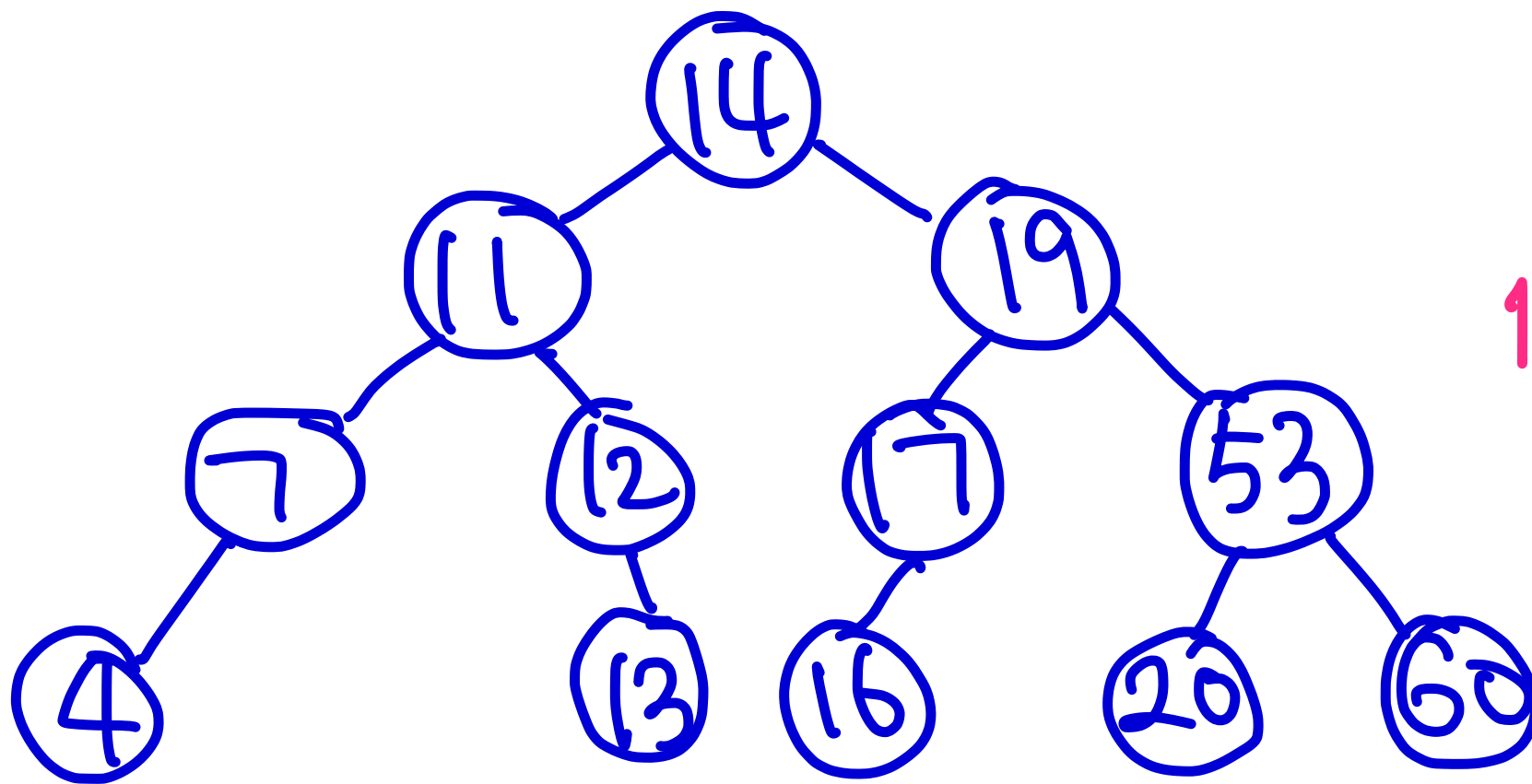
but, 제거할 때마다
balance factor 체크.



여기서 8 제거

1) BT와 같은
방향을 제거

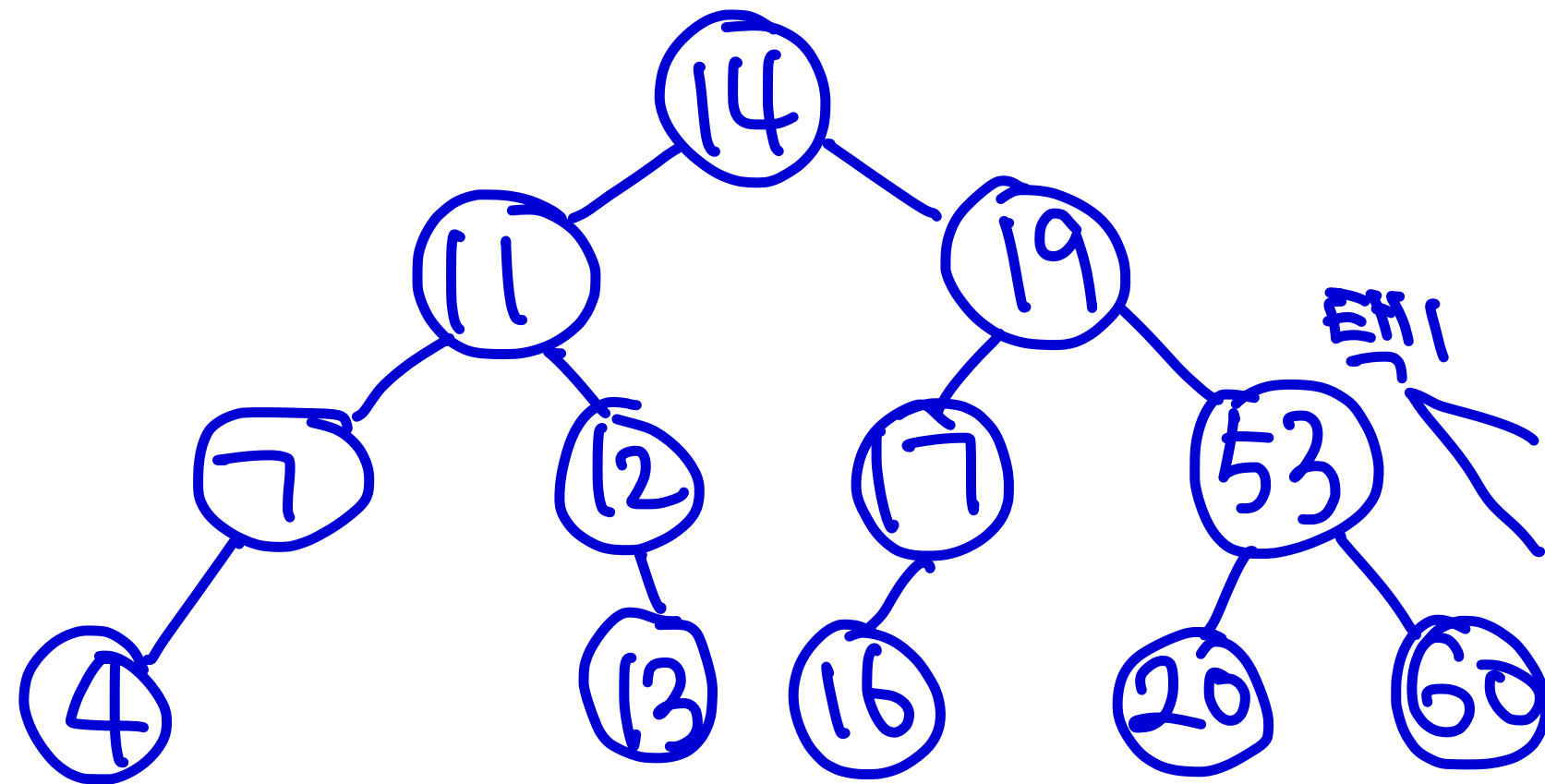
2) 제거할 때마다
bf 체크



여기서 8 제거

1) BT와 같은
방향을 제거

2) 제거할 때마다
bf 체크



서(가)서 7제거

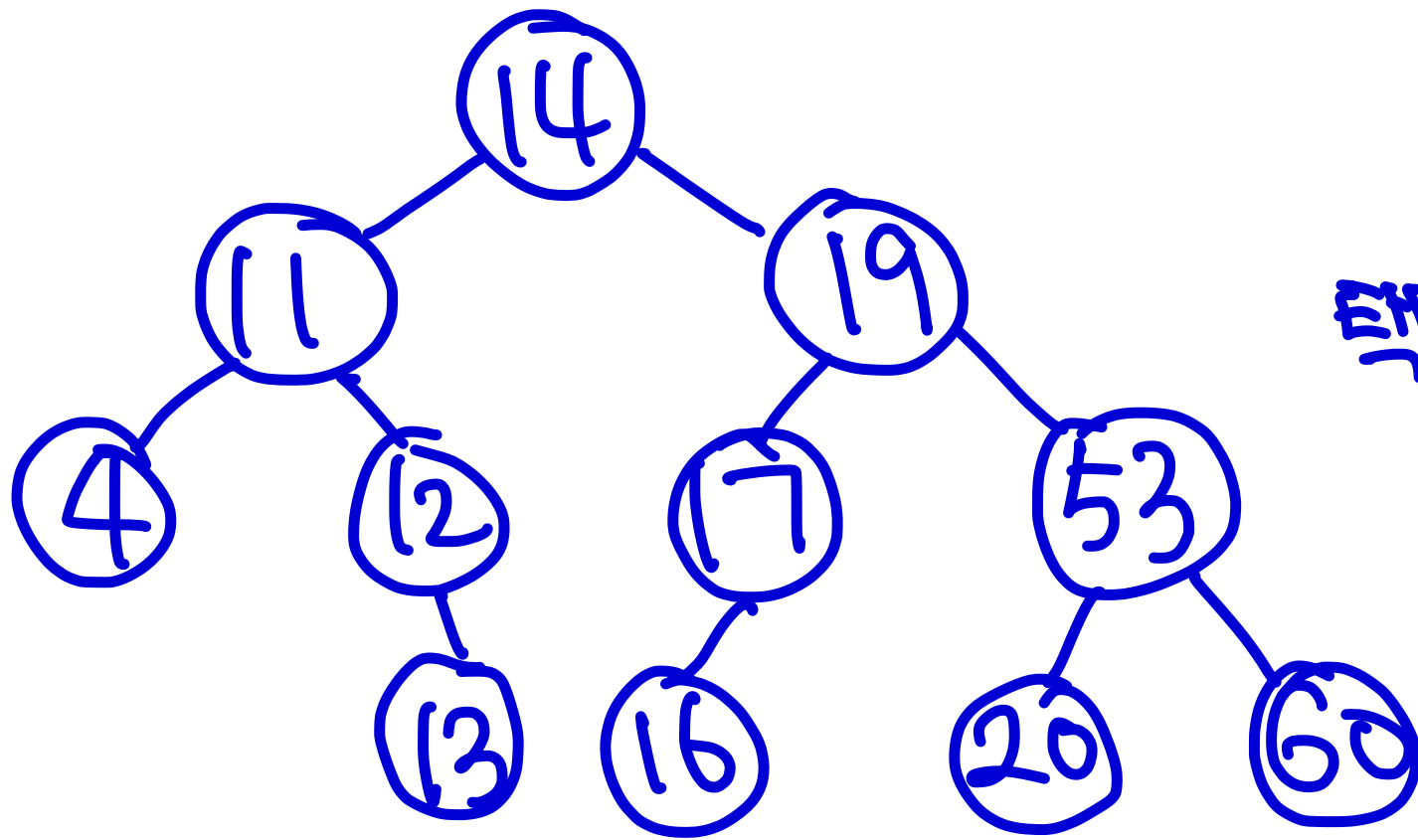
1) BST ~ 제거

InOrder Pred()

InOrder Succ()

2) 서(가)를 찾아

bf 체크



이거 7계거

1) BST ~ 계거

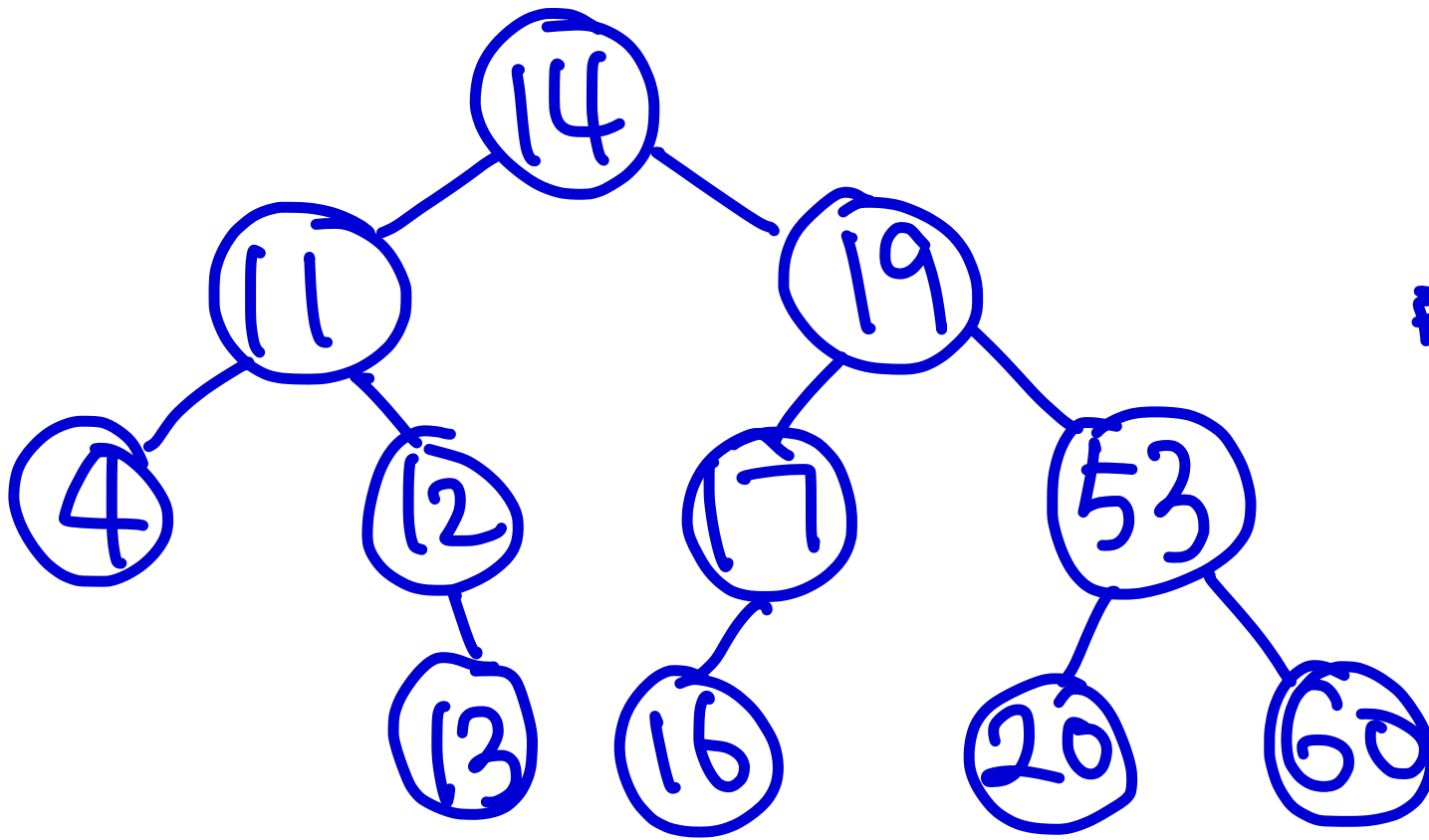
예1

InOrder Pred()

InOrder Succ()

2) 계거 ~ 아니나

bf 체크

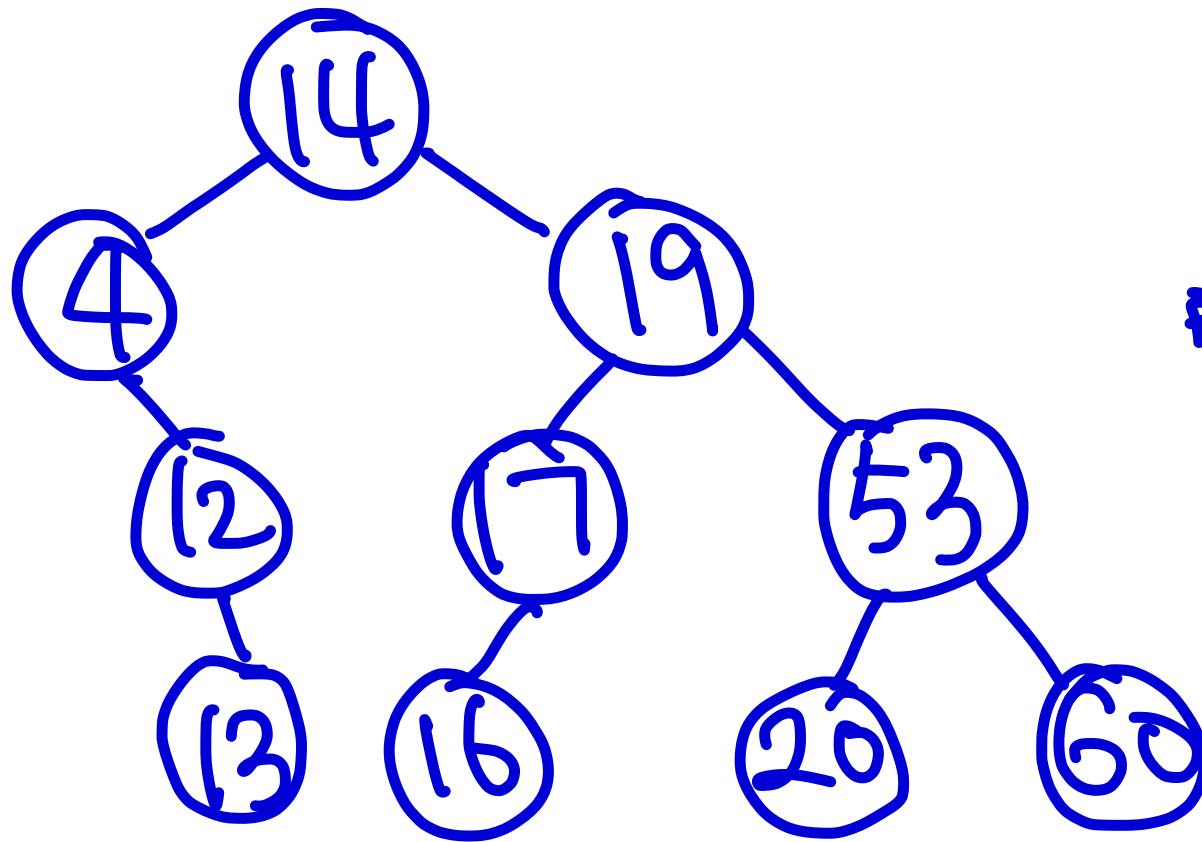


문제 11 제기

예제 1
(1) BST ~ 제기
InOrder Pred()
InOrder Succ()

2) bf 체크

BF가 깨졌으면
왜 깨졌나?



문제 11 제기

예제 1

1) BST ~ 제기

InOrder Pred()

InOrder Succ()

2) bf 체크

BF가 깨졌으면
태가 깨졌나?

3) 회전

① Rotation 대상

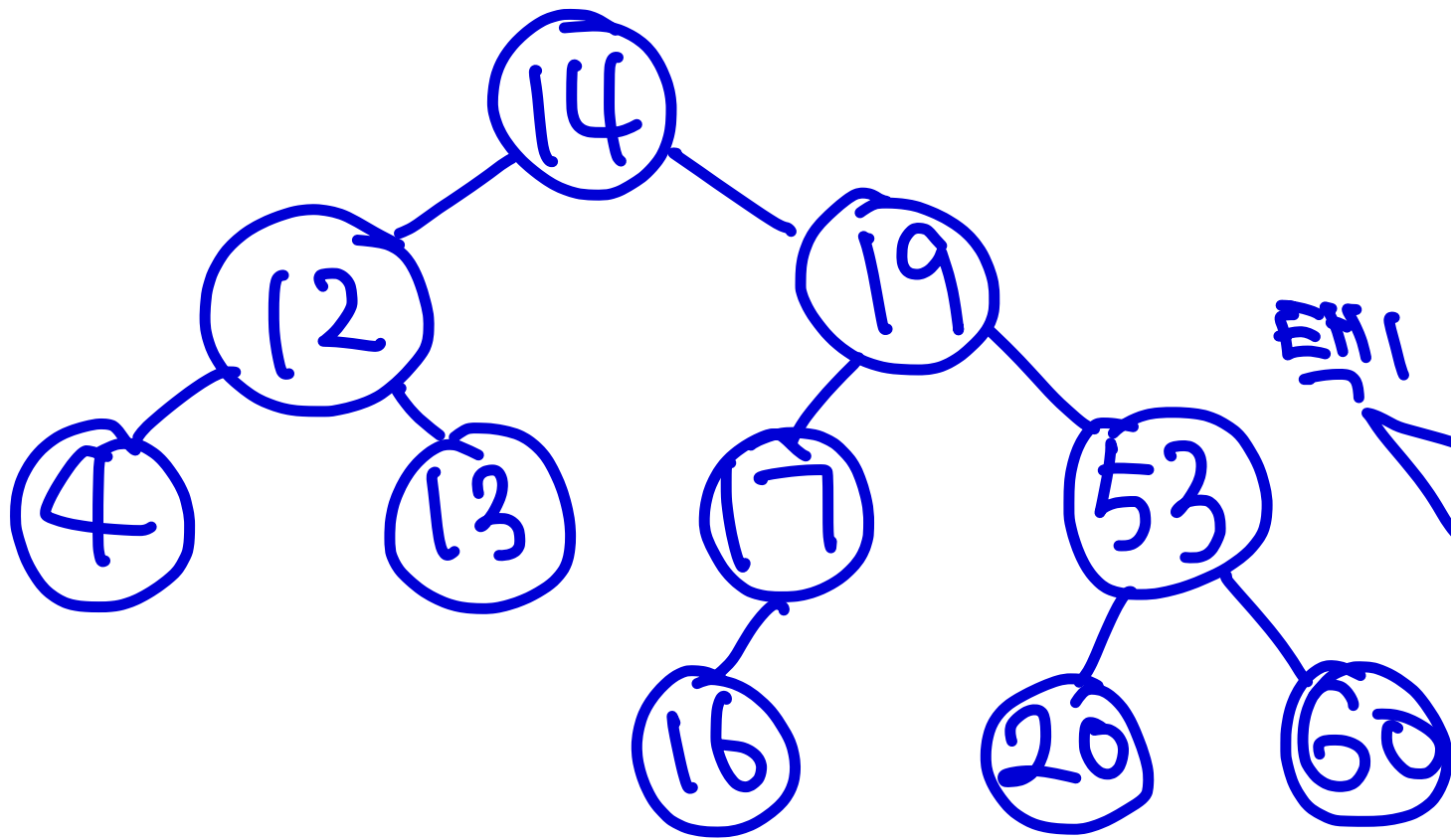
서 노드의 중앙값을 루트로

② 나머지 노드를 재배치

4) 회전 완료 후

전체 bf 다시 확인

AVL tree인가?

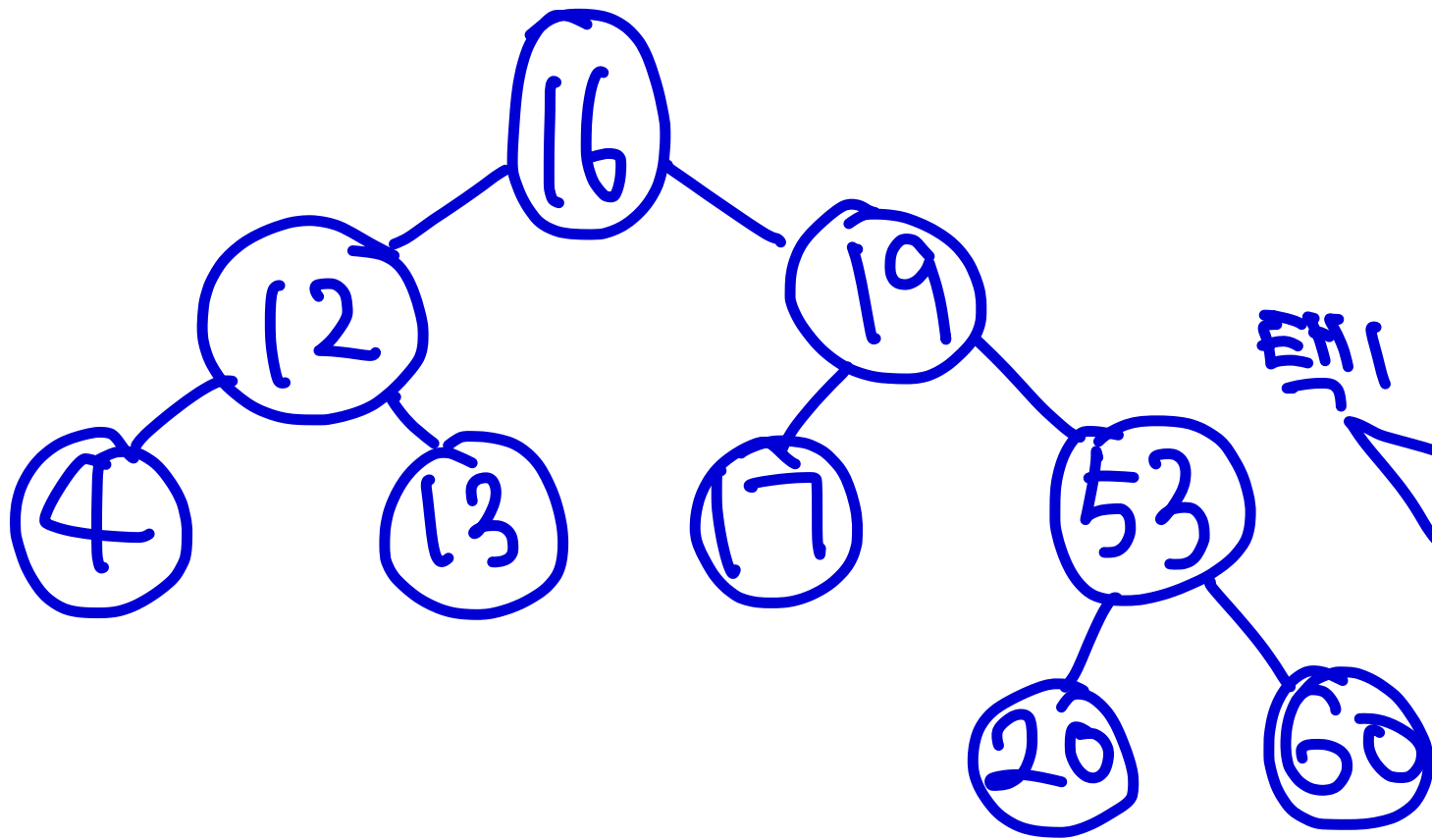


여기서 4 삭제

예1 1) BST 체크
InOrder Pred()
InOrder Succ()

2) bf 체크

BF가 깨졌으면
태깅했나?

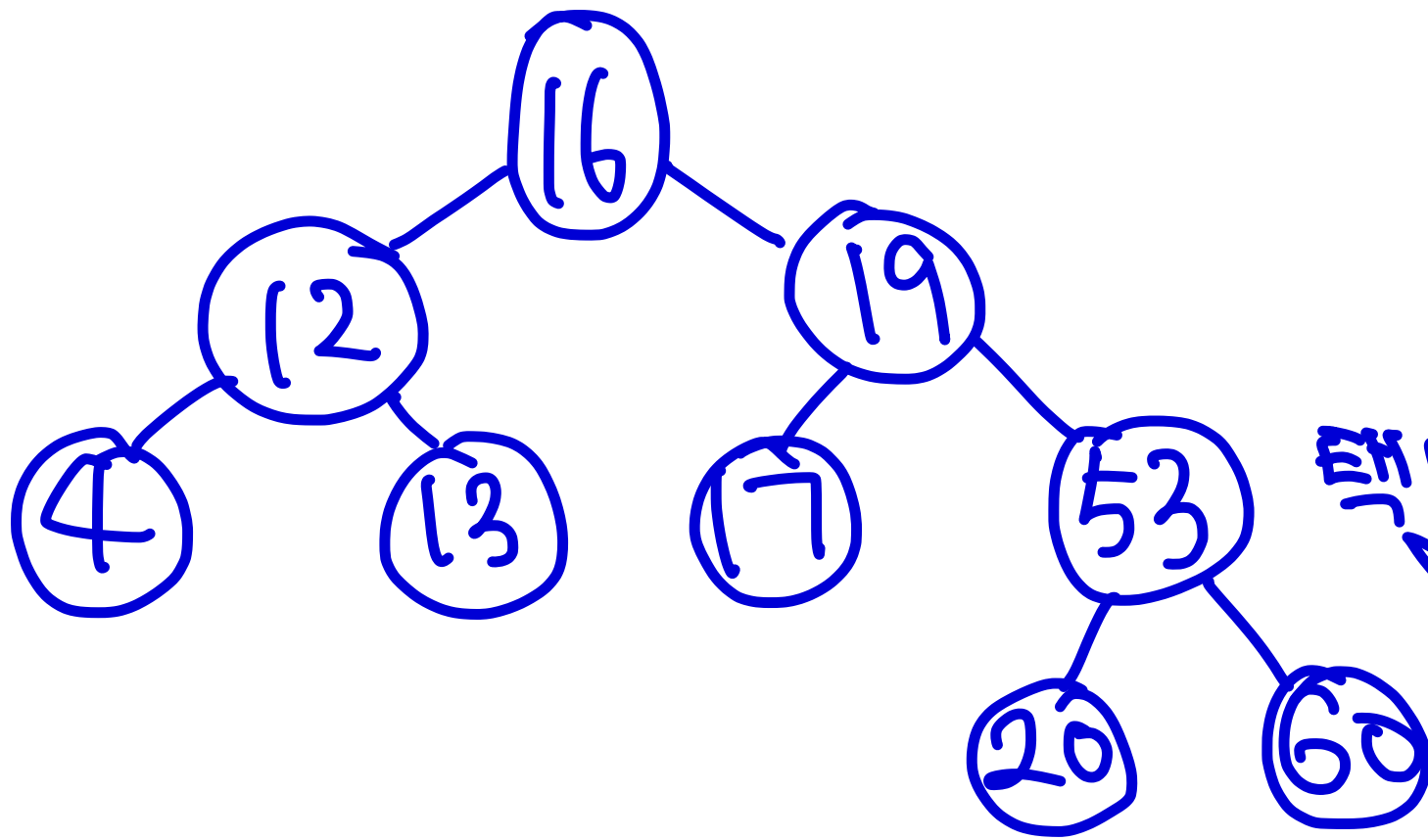


여기서 4 삭제

1) BST 체크
InOrder Pred()
InOrder Succ()

2) bf 체크

BF가 깨졌으면
다들 깨졌나?

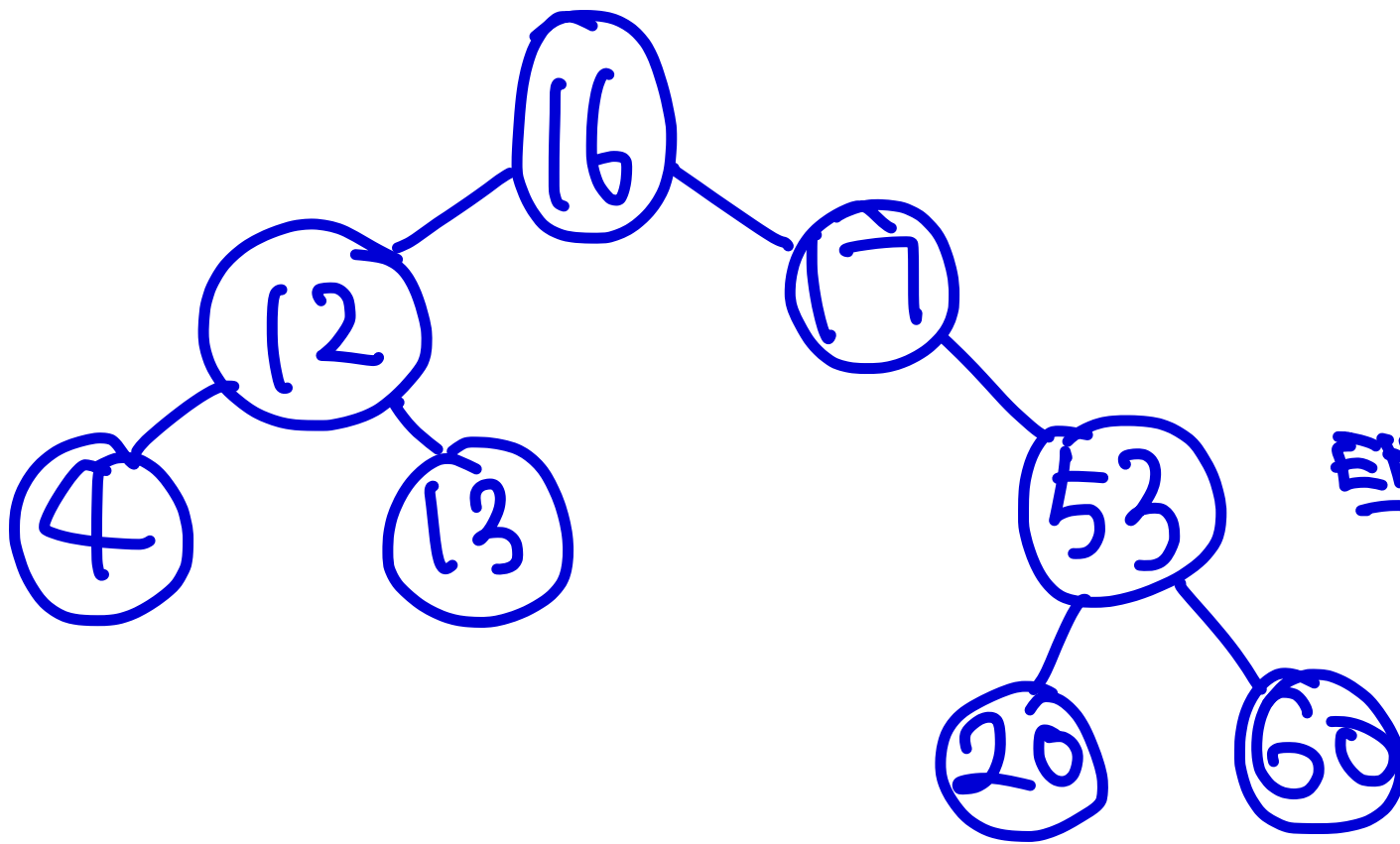


여기서 19 삭제
1) BST ~ 체크

예1
InOrder Pred()
InOrder Succ()

2) bf 체크

BF가 깨졌으면
다들 깨졌나?



여기서 9 삭제
1) BST ~ 체크

예1

InOrder Pred()

InOrder Succ()

2) bf 체크

BF가 깨졌으면

재가장식나?

Case 1 17, 53, 20으로 잡은 경우

들중 택1

3) 회전

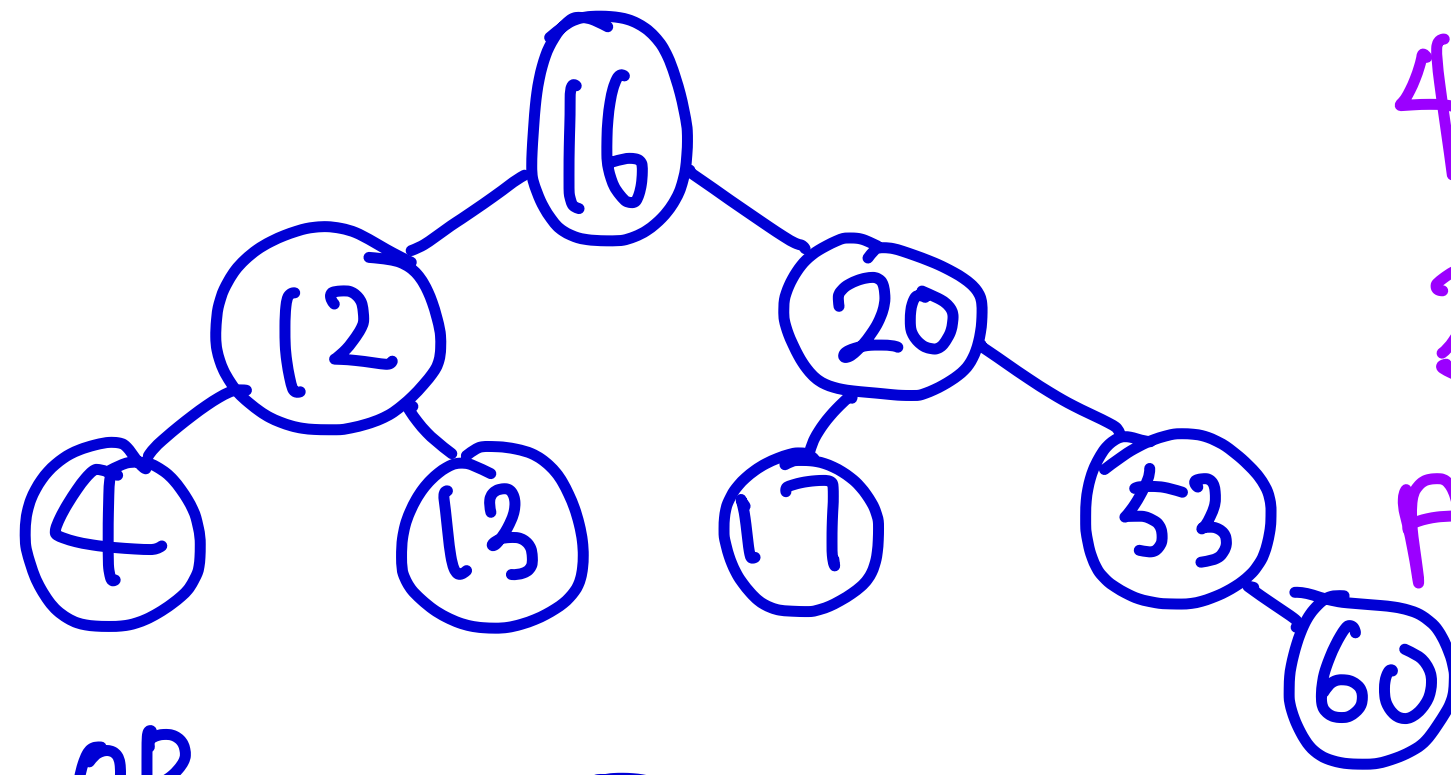
① Rotation 대상

서 노드의 중앙값을 두드려

② 나머지 노드를 재배치

Case 2 17, 53, 60으로 잡은 경우

4) 회전 완료 후
전체 bf 다시 확인
AVL tree인가?



OR

