**<AVL tree>**

**정의**: 모든 노드에서 왼쪽 서브트리의 높이와 오른쪽 서브트리의 높이 차가 1 이하인 이진 탐색 트리를 말한다. -> 한마디로 균형이 잡힌(balanced) 트리 라고 할 수 있다.

**특징**: -트리가 불균형 상태로 되면 스스로 노드들을 재배치하여 균형 상태 유지

-평균, 최선, 최악 시간 복잡도: O(logn)

트리가 불균형 상태가 되면 시간 복잡도는 O(n)이므로 균형으로 만들어야 시간 복잡도가 줄어든 좋은 트리가 된다.

그렇다면 AVL 트리가 **균형이 깨지는 경우**는 어떤게 있을까?

-4가지 경우가 존재한다.

• LL 타입: N이 A의 왼쪽서브트리의 왼쪽서브트리에 삽입

• LR 타입: N이 A의 왼쪽서브트리의 오른쪽서브트리에 삽입

• RR 타입: N이 A의 오른쪽서브트리의 오른쪽서브트리에 삽입

• RL 타입: N이 A의 오른쪽서브트리의 왼쪽서브트리에 삽입

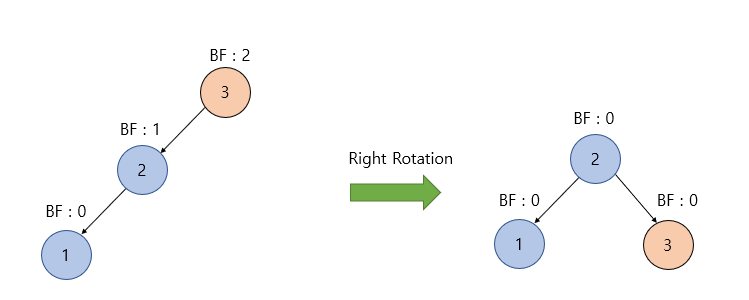
* 각 타입별 **재균형 방법**은?

• LL 회전: A부터 N까지의 경로상 노드의 오른쪽 회전

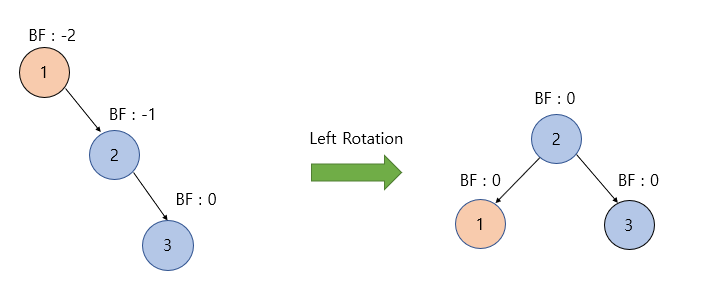
• LR 회전: A부터 N까지의 경로상 노드의 왼쪽-오른쪽 회전

• RR 회전: A부터 N까지의 경로상 노드의 왼쪽 회전

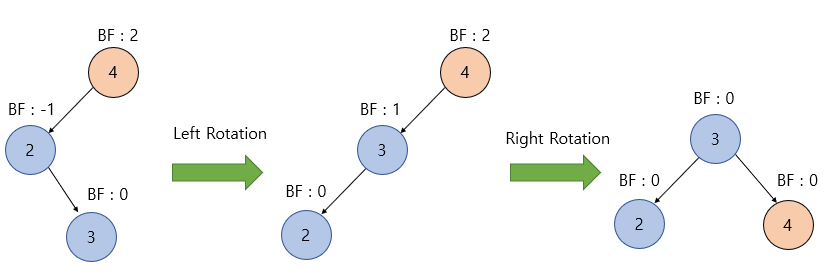
• RL 회전: A부터 N까지의 경로상 노드의 오른쪽-왼쪽 회전



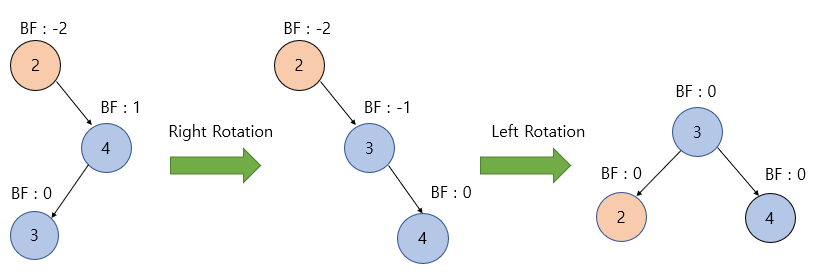
<LL 회전>



<RR 회전>



<LR 회전>



<RL 회전>

AVL 트리를 생성하는 예는 다음과 같다.

