

# 1 | 이진 탐색 트리의 탐색, 삽입, 삭제 연산

## 1 이진 탐색 트리(Binary Search Tree)

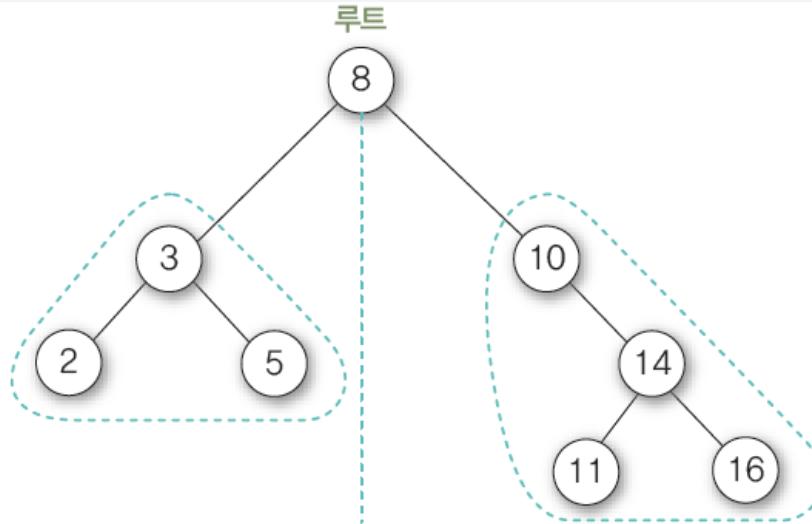
▶ 이진 트리를 탐색용 자료구조로 사용하기 위해 원소 크기에 따라 노드 위치를 정의한 것

- 모든 원소는 서로 다른 유일한 키를 갖는다.
- 왼쪽 서브 트리에 있는 원소들의 키는 그 루트의 키보다 작다.
- 오른쪽 서브 트리에 있는 원소들의 키는 그 루트의 키보다 크다.
- 왼쪽 서브 트리와 오른쪽 서브 트리도 이진 탐색 트리이다.

### [이진 탐색 트리의 정의]

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 1 이진 탐색 트리(Binary Search Tree)



왼쪽 서브 트리의 키값 < 루트의 키값 < 오른쪽 서브 트리의 키값

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 2 이진 탐색 트리의 탐색 연산

- ▶ 루트에서 시작
- ▶ 탐색할 키값  $x$ 를 루트 노드의 키값과 비교
  - (키값  $x =$  루트 노드의 키값)인 경우  
: 원하는 원소를 찾았으므로 탐색연산 성공
  - (키값  $x <$  루트 노드의 키값)인 경우  
: 루트노드의 왼쪽 서브트리에 대해서  
탐색연산 수행
  - (키값  $x >$  루트 노드의 키값)인 경우  
: 루트노드의 오른쪽 서브트리에 대해서  
탐색연산 수행
- ▶ 서브트리에 대해서 순환적으로 탐색 연산을 반복

## 2 이진 탐색 트리의 탐색 연산

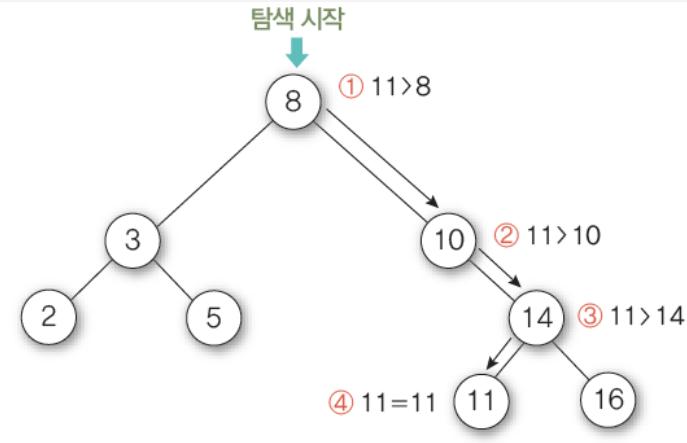
알고리즘 7-4 이진 탐색 트리의 노드 탐색

```
searchBST(bsT, x)
    p ← bsT;
    if (p = NULL) then
        return NULL;
    if (x = p.key) then
        return p;
    if (x < p.key) then
        return searchBST(p.left, x);
    else return searchBST(p.right, x);
end searchBST()
```

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 2 이진 탐색 트리의 탐색 연산

### 1) 이진 탐색 트리에서 원소 11을 탐색



- ① (찾는 키값 11 > 루트 노드의 키값 8)이므로 오른쪽 서브 트리 탐색
- ② (찾는 키값 11 > 노드의 키값 10)이므로 다시 오른쪽 서브 트리 탐색
- ③ (찾는 키값 11 < 노드의 키값 14)이므로 왼쪽 서브 트리 탐색
- ④ (찾는 키값 11 = 노드의 키값 11)이므로 탐색 성공으로 연산 종료

### [이진 탐색 트리에서 탐색 과정 예]

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 3 이진 탐색 트리의 삽입 연산

- ① 먼저 탐색 연산을 수행
  - ▶ 삽입할 원소와 같은 원소가 트리에 있으면  
삽입할 수 없으므로, 같은 원소가 트리에 있는지  
탐색하여 확인
  - ▶ 탐색에서 탐색 실패가 결정되는 위치가 삽입 위치가 됨
- ② 탐색 실패한 위치에 원소를 삽입

## 3 이진 탐색 트리의 삽입 연산

### 1) 이진 탐색 트리에서 삽입 연산을 하는 알고리즘

- ▶ 삽입할 자리를 찾기 위해 포인터 p를 사용,  
삽입할 노드의 부모 노드를 지정하기 위해 포인터 q를 사용

```
알고리즘 7-5 이진 탐색 트리의 노드 삽입

insertBST(bsT, x)
    p ← bsT
    while (p ≠ NULL) do {
        if (x = p.key) then return;
        q ← p;
        if (x < p.key) then p ← p.left;
        else p ← p.right;
    }

    new ← getNode();
    new.key ← x;
    new.left ← NULL;
    new.right ← NULL;

    if (bsT = NULL) then bsT ← new;
    else if (x < q.key) then q.left ← new;
    else q.right ← new;
    return;
end insertBST()
```

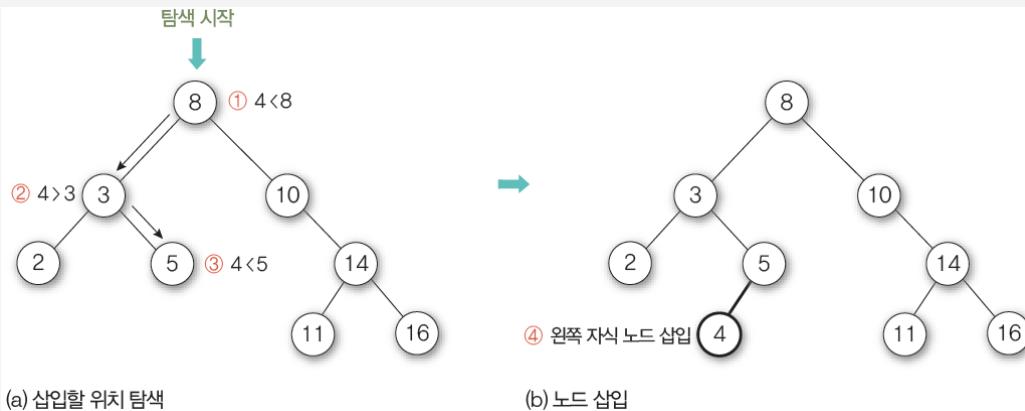
The diagram illustrates the execution flow of the `insertBST` algorithm. It is divided into three main sections, each enclosed in a brace and labeled with a circled number:

- ① 삽입할 노드 탐색**: This section covers the first part of the algorithm where the insertion node is found by traversing the tree starting from the root (`bsT`). The loop continues until a node with the same key is found or the current node is null.
- ② 삽입할 노드 생성**: This section covers the creation of the new node that is to be inserted. It uses the `getNode` function to allocate memory for the new node and initializes its key to `x`, and its left and right pointers to `NULL`.
- ③ 삽입 노드 연결**: This section covers the final step of connecting the new node to the tree. It handles three cases:
  - If the current node (`bsT`) is null, it becomes the new node.
  - If the new node's key is less than the current node's key, it becomes the left child of the current node.
  - If the new node's key is greater than the current node's key, it becomes the right child of the current node.After the connection is made, the function returns.

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 3 이진 탐색 트리의 삽입 연산

### 2) 이진 탐색 트리에 원소 4를 삽입 하기



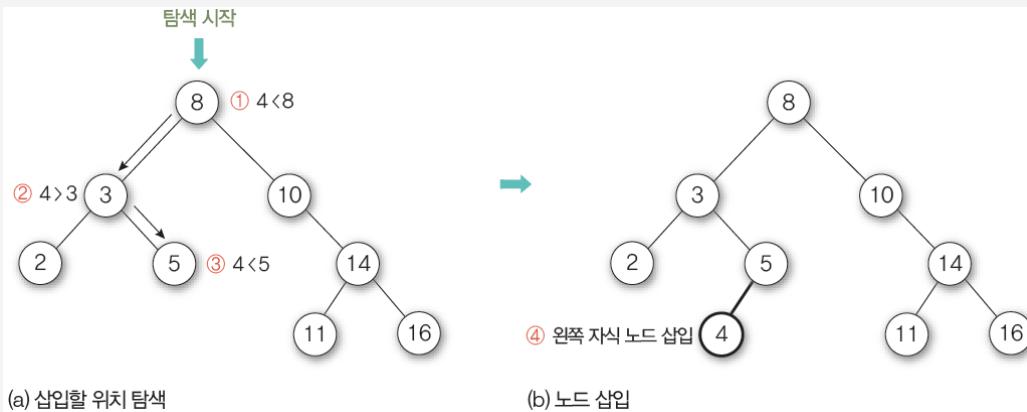
① (찾는 키값  $4 <$  루트 노드의 키값 8)이므로 왼쪽 서브 트리를 탐색

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

# 1 | 이진 탐색 트리의 탐색, 삽입, 삭제 연산

## 3 이진 탐색 트리의 삽입 연산

### 2) 이진 탐색 트리에 원소 4를 삽입 하기



(a) 삽입할 위치 탐색

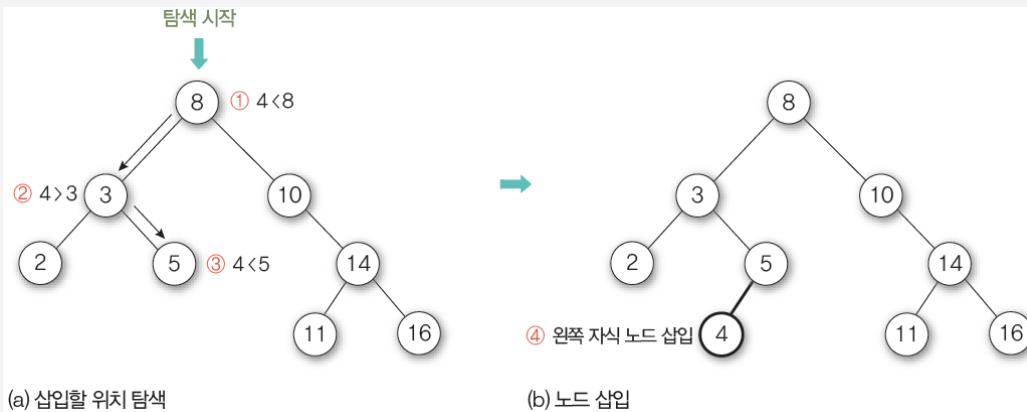
(b) 노드 삽입

② (찾는 키값 4 > 노드의 키값 3)이므로 오른쪽 서브 트리를 탐색

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 3 이진 탐색 트리의 삽입 연산

### 2) 이진 탐색 트리에 원소 4를 삽입 하기



(a) 삽입할 위치 탐색

(b) 노드 삽입

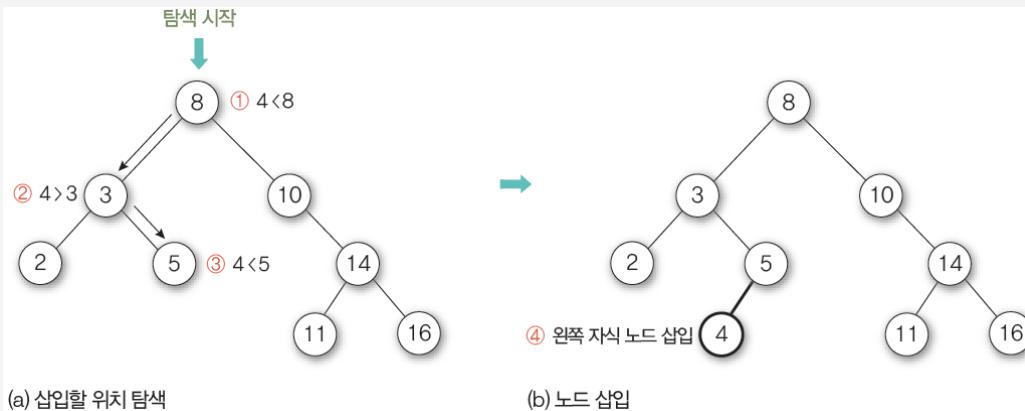
③ (찾는 키값  $4 <$  노드의 키값 5)이므로 왼쪽 서브 트리를 탐색해야 하지만,  
왼쪽 자식 노드가 없으므로 노드 5의 왼쪽 자식 노드에서 탐색 실패가 발생

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

# 1 | 이진 탐색 트리의 탐색, 삽입, 삭제 연산

## 3 이진 탐색 트리의 삽입 연산

### 2) 이진 탐색 트리에 원소 4를 삽입 하기

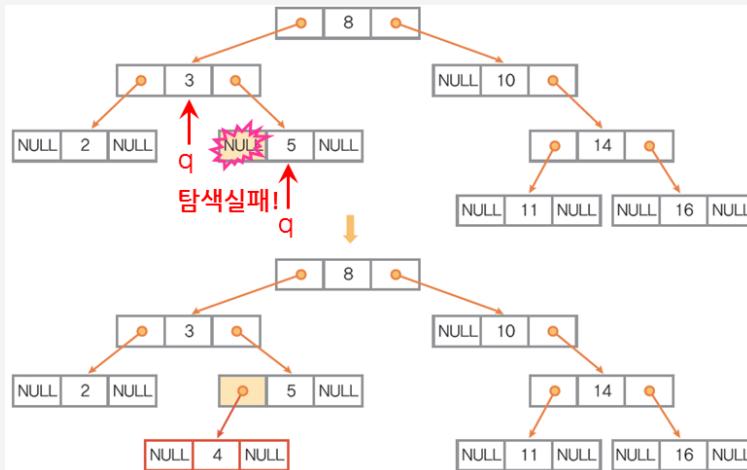


④ 실패가 발생한 자리, 즉 노드 5의 왼쪽 자식 노드 자리에 노드 4를 삽입

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 3 이진 탐색 트리의 삽입 연산

3) 이진 탐색 트리에 원소 4를 삽입하는 과정을  
연결 자료구조로 표현



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 4 이진 탐색 트리의 삭제 연산

- ① 먼저 탐색 연산을 수행
  - ▶ 삭제할 노드의 위치를 알아야 하므로 트리를 탐색

## 4 이진 탐색 트리의 삭제 연산

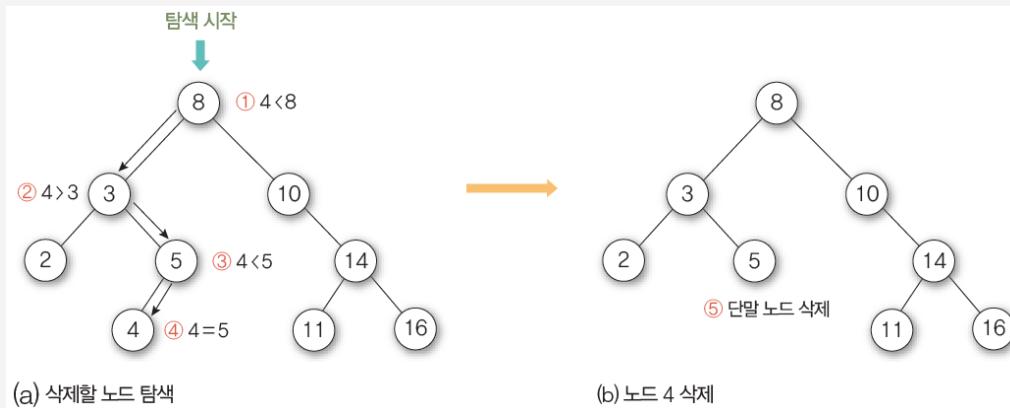
### ② 탐색 성공한 위치에 원소를 삭제

- ▶ 노드의 삭제 후에도 이진 탐색 트리를 유지해야 하므로  
삭제 노드의 경우에 대한 후속 처리(이진 탐색  
트리의 재구성 작업)가 필요함
  - 삭제할 노드의 경우
    - ✓ 삭제할 노드가 단말 노드인 경우(차수 = 0)
    - ✓ 삭제할 노드가 자식 노드를 한 개 가진 경우  
(차수 = 1)
    - ✓ 삭제할 노드가 자식 노드를 두 개 가진 경우  
(차수 = 2)

## 4 이진 탐색 트리의 삭제 연산

1) 삭제할 노드가 단말 노드인 경우(차수 =0)의 삭제 연산

## ▶ 노드 4를 삭제하는 경우

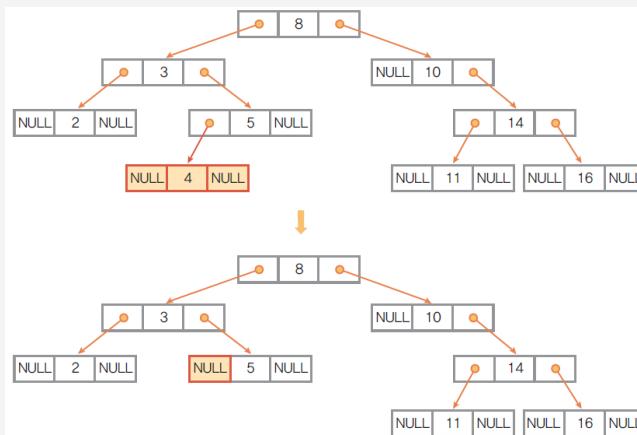


\*출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 4 이진 탐색 트리의 삭제 연산

1) 삭제할 노드가 단말 노드인 경우(차수 =0)의 삭제 연산

▶ 4를 삭제하기 전과 후로 연결 자료구조를 표현



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

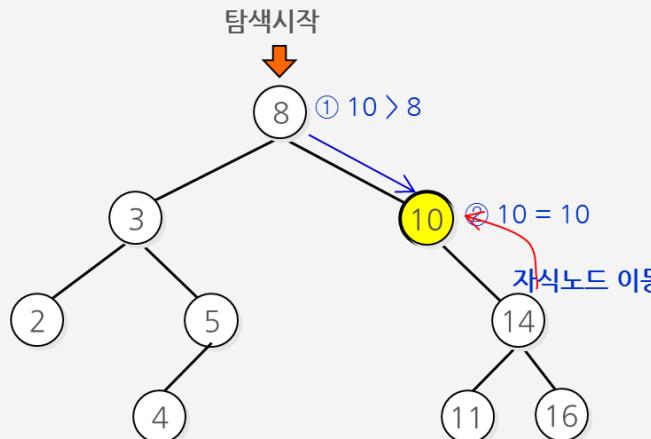
## 4 이진 탐색 트리의 삭제 연산

2) 삭제할 노드가 자식 노드를 한 개 가진 경우(차수 =1)의  
삭제 연산

- ▶ 노드를 삭제하면, 자식 노드는 트리에서 연결이 끊어져서 고아가 됨
- ▶ 후속 처리 : 삭제한 부모노드의 자리를 자식노드에게 물려줌
- ▶ 노드 10을 삭제하는 경우

## 4 이진 탐색 트리의 삭제 연산

2) 삭제할 노드가 자식 노드를 한 개 가진 경우(차수 =1)의  
삭제 연산



1단계 : 삭제할 노드 **탐색**

2단계 : 탐색한 노드 **삭제**

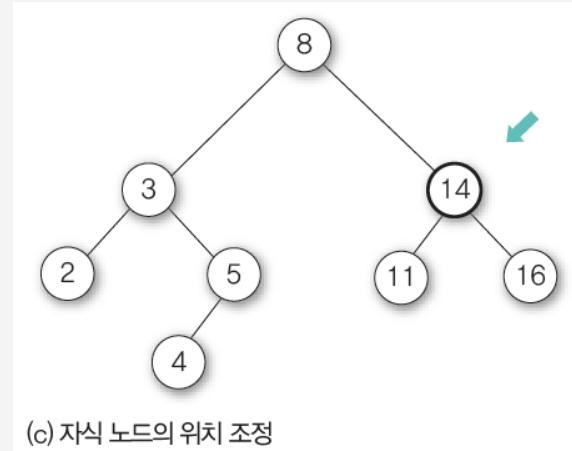
3단계 : **후속처리**

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 4 이진 탐색 트리의 삭제 연산

2) 삭제할 노드가 자식 노드를 한 개 가진 경우(차수 =1)의  
삭제 연산

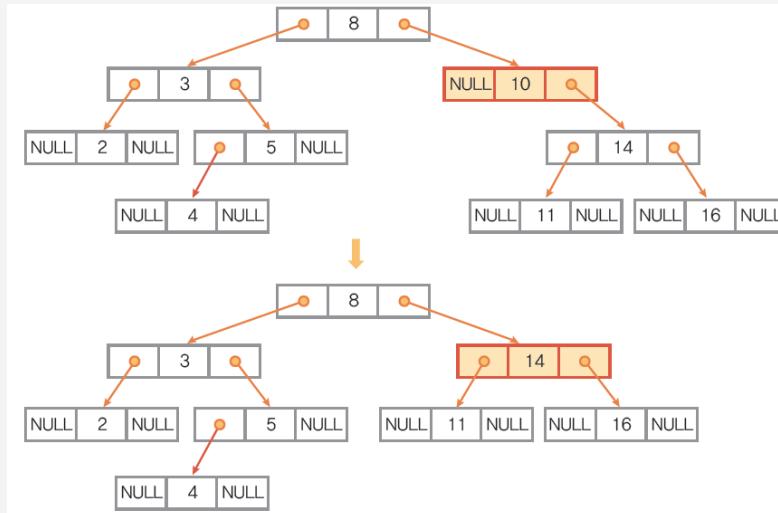
- ▶ 노드 10을 삭제  
: 삭제한 부모노드의  
10자리를 자식노드  
14에게 물려줌



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 4 이진 탐색 트리의 삭제 연산

### 3) 노드 10을 삭제하는 경우에 대한 단순 연결 리스트 표현



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 4 이진 탐색 트리의 삭제 연산

4) 삭제할 노드가 자식 노드를 두 개 가진 경우(차수 =2)의  
삭제 연산

- ▶ 노드를 삭제하면, 자식 노드들은 트리에서 연결이 끊어져서 고아가 됨
- ▶ 후속 처리 : 삭제한 노드의 자리를 자손 노드들 중에서 선택한 후계자에게 물려줌

## 4 이진 탐색 트리의 삭제 연산

4) 삭제할 노드가 자식 노드를 두 개 가진 경우(차수 =2)의  
삭제 연산

### ▶ 후계자 선택 방법

- 왼쪽 서브트리에서 가장 큰 자손노드 선택 :  
왼쪽 서브트리의 오른쪽 링크를 따라  
계속 이동하여 오른쪽 링크 필드가 NULL인 노드  
즉, 가장 오른쪽 노드가 후계자가 됨

## 4 이진 탐색 트리의 삭제 연산

4) 삭제할 노드가 자식 노드를 두 개 가진 경우(차수 =2)의  
삭제 연산

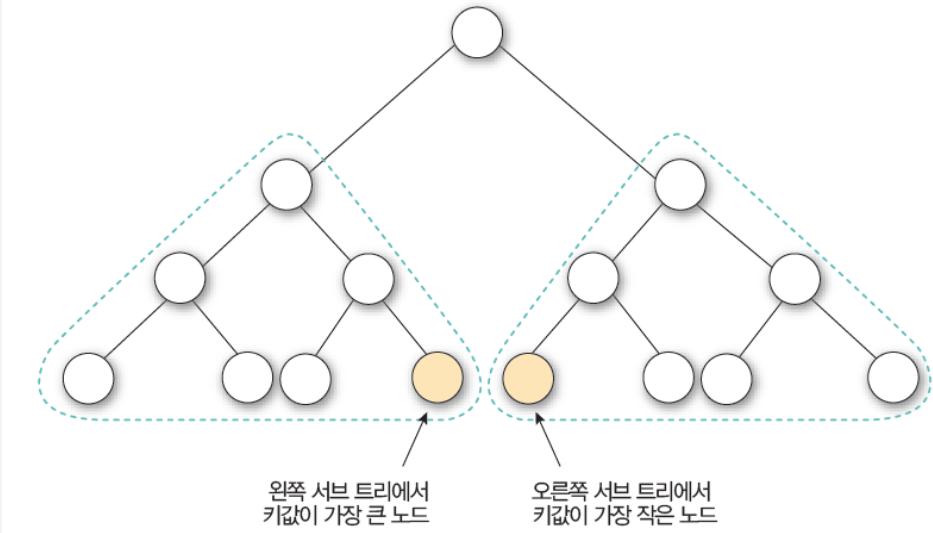
### ▶ 후계자 선택 방법

- 오른쪽 서브트리에서 가장 작은 자손노드 선택 :  
오른쪽 서브트리에서 왼쪽 링크를 따라  
계속 이동하여 왼쪽 링크 필드가 NULL인 노드  
즉, 가장 왼쪽에 있는 노드가 후계자가 됨

## 4 이진 탐색 트리의 삭제 연산

4) 삭제할 노드가 자식 노드를 두 개 가진 경우(차수 =2)의  
삭제 연산

- ▶ 삭제한 노드의 자리를  
물려받을 수 있는 자손 노드

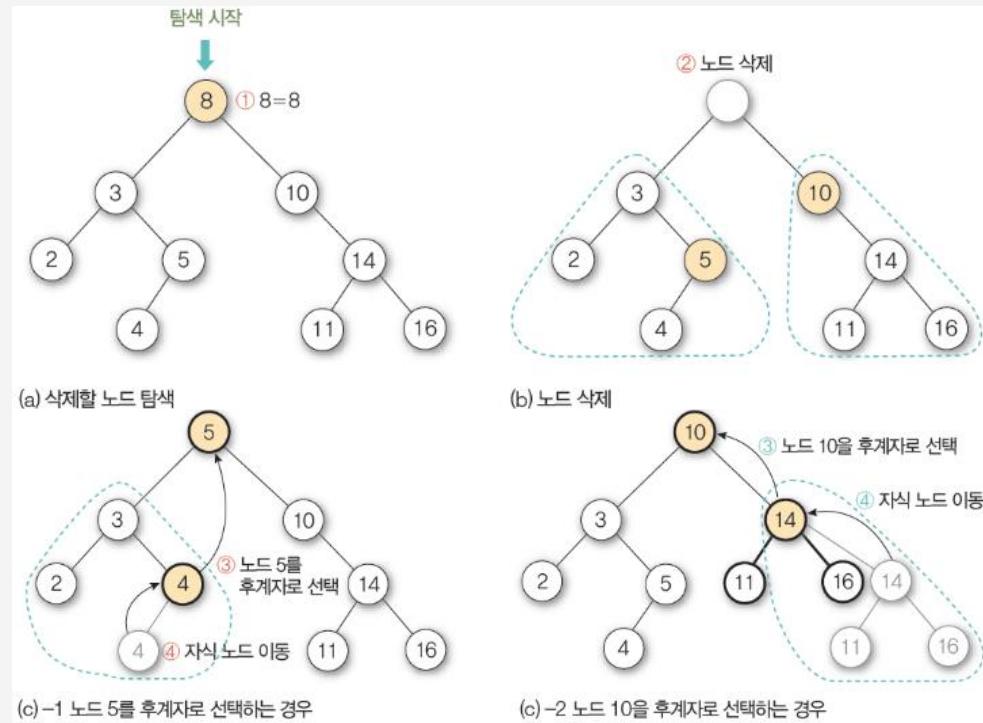


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 4 이진 탐색 트리의 삭제 연산

### 5) 노드 8을 삭제하는 경우

[이진 탐색 트리에서 자식 노드가  
둘인 노드 8을 삭제하는 예]



※출처: 이지영(2016). IT CookBook,  
C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 4 이진 탐색 트리의 삭제 연산

### 5) 노드 8을 삭제하는 경우

#### ▶ 노드 5를 후계자로 선택한 경우

- ① 후계자 노드 5를 원래 자리에서 삭제하여,  
삭제노드 8의 자리를 물려줌
- ② 후계자 노드 5의 원래 자리는 자식노드 4에게  
물려주어 이진 탐색 트리를 재구성  
(자식노드가 하나인 노드 삭제 연산의  
후속처리 수행)

## 4 이진 탐색 트리의 삭제 연산

### 5) 노드 8을 삭제하는 경우

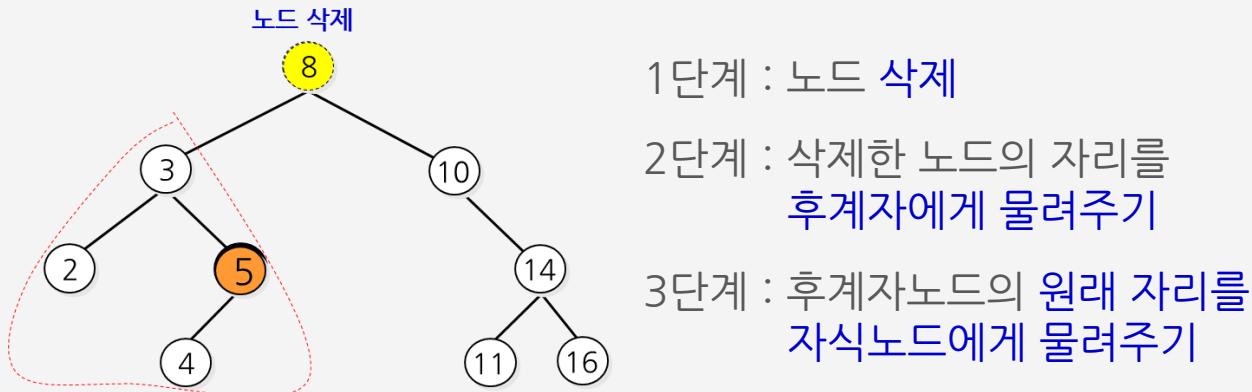
#### ▶ 노드 10을 후계자로 선택한 경우

- ① 후계자 노드 10을 원래 자리에서 삭제하여,  
삭제노드 8의 자리를 물려줌
- ② 후계자 노드 10의 원래 자리는 자식노드 14에게  
물려주어 이진 탐색 트리를 재구성  
(자식노드가 하나인 노드 삭제 연산의  
후속처리 수행)

## 4 이진 탐색 트리의 삭제 연산

5) 노드 8을 삭제하는 경우

▶ 노드 5를 후계자로 선택한 경우

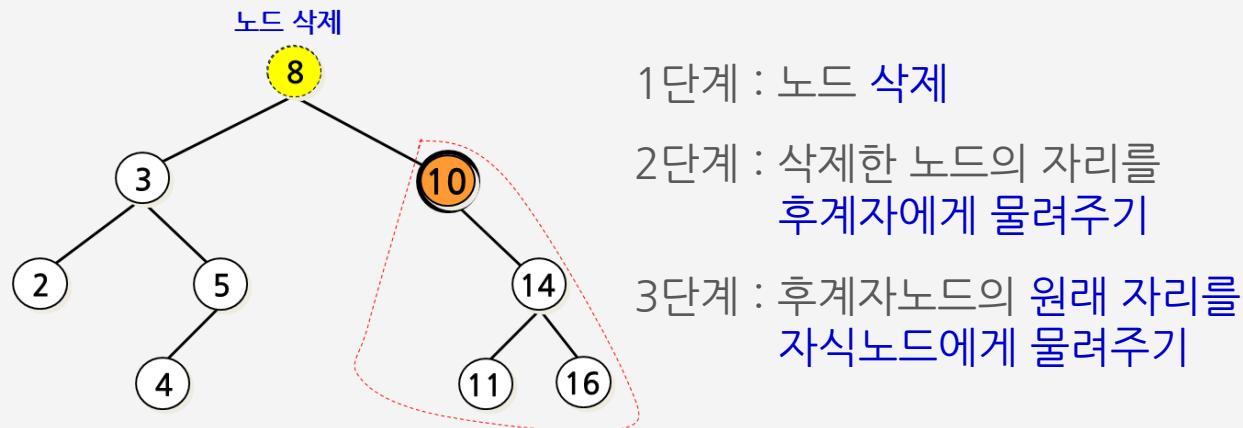


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 4 이진 탐색 트리의 삭제 연산

5) 노드 8을 삭제하는 경우

▶ 노드 10를 후계자로 선택한 경우



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 2 | 균형 이진 탐색 트리와 AVL 트리의 개념

### 1 균형 이진 탐색 트리의 개념

- ▶ 이진탐색 트리에서 좌우 균형이 잘 맞으면 탐색의 성능이 높고, 이 성능은 탐색 트리의 높이와 밀접한 상관이 있음
- ▶ 이진 탐색 트리에 왼쪽 서브 트리의 높이와 오른쪽 서브 트리의 높이에 대한 균형 조건을 추가하여 정의한 트리를 균형 이진 탐색 트리(Balanced Binary Search Tree) 또는 균형 트리(Balanced Tree)라고 함

### 2 AVL 트리의 개념과 유형

- ▶ AVL 트리(Adelson-Velskii, Landis Tree)는 대표적인 균형 이진 탐색 트리
- ▶ 각 노드에서 왼쪽 서브 트리의 높이  $hL$ (Height of Left subtree)과 오른쪽 서브 트리의 높이  $hR$ (Height of Right Subtree)의 차이가 1 이하인 트리

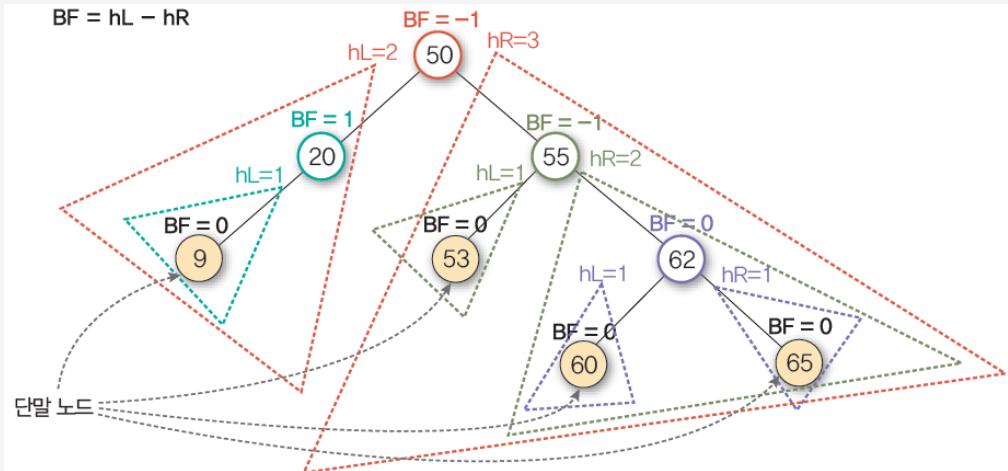
### 2 AVL 트리의 개념과 유형

#### ▶ 특징

- 왼쪽 서브 트리 < 부모 노드 < 오른쪽 서브 트리의 크기 관계를 갖음
- 각 노드의 왼쪽 서브 트리 높이와 오른쪽 서브 트리 높이의 차이( $h_L - h_R$ )를 노드의 균형 인수 (BF, Balance Factor)라 함
- 각 노드의 균형 인수로 {-1, 0, +1} 값만 가지게 함으로써 왼쪽 서브 트리와 오른쪽 서브 트리의 균형을 항상 유지함

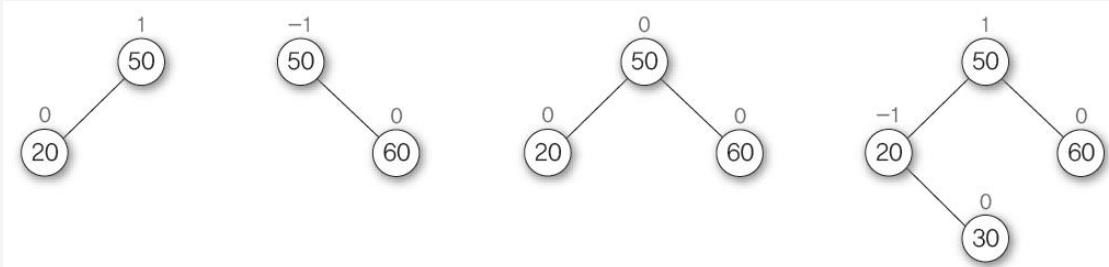
### 3 AVL 트리의 균형인수

- ▶ 균형 인수 BF는 각 노드에서 왼쪽 서브 트리의 높이  $h_L$ 과 오른쪽 서브 트리의 높이  $h_R$ 의 차이



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

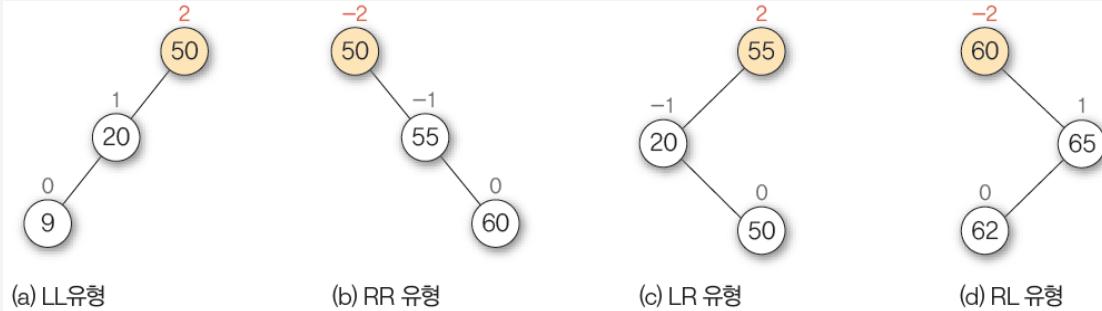
### 4 AVL 트리의 예



- ▶ 모든 노드가 균형 인수로 {-1, 0, 1} 값을 가지므로 AVL 트리임

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

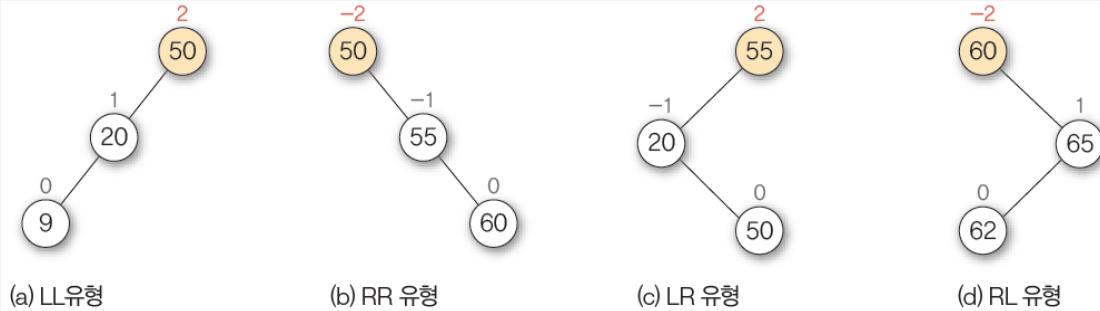
### 5 비AVL 트리의 예



- ▶ 노드들이 균형 인수로  $\{-1, 0, 1\}$  외의 값을 가지므로 균형이 깨져 한 방향으로 치우친 비AVL 트리임

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

### 5 비AVL 트리의 예



- ▶ 균형이 깨진 노드의 균형 인수가 +이면 왼쪽 서브 트리에 문제가 있는 것이고, 균형 인수가 -이면 오른쪽 서브 트리에 문제가 있음

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

### 6 AVL 트리에서 발생하는 불균형

#### ▶ LL 유형(Left-Left 유형)

- (a)는 노드 50의 균형 인수가 +2로 균형이 깨진 노드인데 균형 인수가 +이므로 왼쪽서브 트리가 문제
- 그리고 노드 50의 왼쪽(L) 자식인 노드 20의 균형 인수가 +이므로 노드 20의 왼쪽(L) 자식 노드가 불균형과 관련 있음
- 따라서 노드 50이 불균형 노드가 된 이유는 노드 50이 왼쪽(L) 자식 노드와 자식의 왼쪽(L) 자식 노드를 가짐으로써 왼쪽으로 치우쳤기 때문임

### 6 AVL 트리에서 발생하는 불균형

#### ▶ RR 유형(Right-Right 유형)

- (b)는 노드 50의 균형 인수가 -2로 균형이 깨진 노드인데 균형 인수가 -이므로 오른쪽 서브 트리에 문제임
- 그리고 노드 50의 오른쪽(R) 자식인 노드 55의 균형 인수가 -이므로 노드 55의 오른쪽(R) 자식 노드가 불균형과 관련 있음
- 따라서 노드 50이 불균형 노드가 된 이유는 노드 50이 오른쪽(R) 자식 노드와 자식의 오른쪽(R) 자식 노드를 가짐으로써 오른쪽으로 치우쳤기 때문임

### 6 AVL 트리에서 발생하는 불균형

#### ▶ LR 유형(Left-Right 유형)

- (c)는 노드 55의 균형 인수가 +2로 균형이 깨진 노드인데 균형 인수가 +이므로 왼쪽 서브 트리에 문제임
- 그리고 노드 55의 왼쪽(L) 자식인 노드 20의 균형 인수는 -이므로 노드 20의 오른쪽(R) 자식 노드가 불균형과 관련 있음
- 따라서 노드 55가 불균형 노드가 된 이유는 왼쪽(L) 자식 노드와 자식의 오른쪽(R) 자식 노드로 인해 왼쪽 서브 트리가 치우쳤기 때문임

### 6 AVL 트리에서 발생하는 불균형

#### ▶ RL 유형(Right-Left 유형)

- (d)는 노드 60의 균형 인수가 -2로 균형이 깨진 노드인데 균형 인수가 -이므로 오른쪽 서브 트리에 문제임
- 그리고 노드 60의 오른쪽(R) 자식인 노드 65의 균형 인수는 +이므로 노드 65의 왼쪽(L) 자식 노드가 불균형과 관련 있음
- 따라서 노드 60이 불균형 노드가 된 이유는 오른쪽(R) 자식 노드와 자식의 왼쪽(L) 자식 노드로 인해 오른쪽 서브 트리가 치우쳤기 때문임

### 3 | AVL 트리의 회전 연산

#### 1 AVL 트리의 회전 연산

- ▶ AVL 트리에서 수행하는 삽입·삭제 작업은 이진 탐색 트리에서의 삽입·삭제 작업과 같고, 이후에 균형을 맞추어주는 재구성 작업이 추가되는데 이 작업은 회전(Rotation) 연산을 통해 이루어짐
  - 단순 회전(Single Rotation)
    - LL 회전과 RR 회전과 같이 한 번 회전하는 것
  - 이중 회전(Double Rotation)
    - LR 회전과 RL 회전과 같이 두 번 회전하는 것

## 1 AVL 트리의 회전 연산

### [AVL 트리에서의 불균형 유형별 해결 방법]

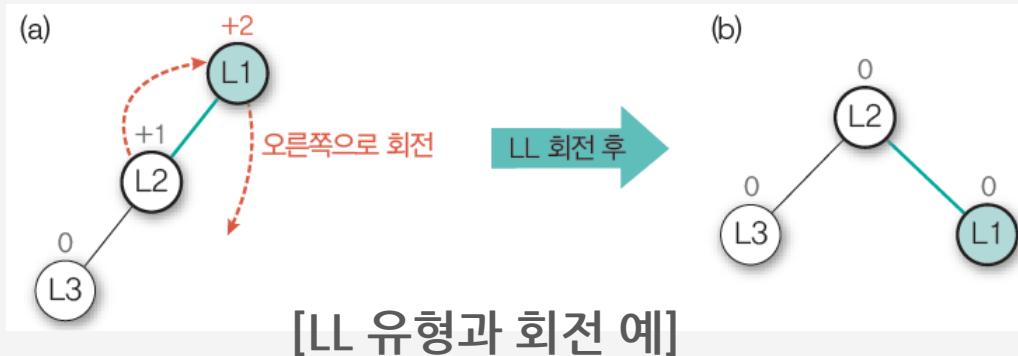
불균형의 유형과 원인	해결 방법
LL 유형 : 불균형 발생 노드의 왼쪽 <sup>Left</sup> 자식 노드와 자식의 왼쪽 <sup>Left</sup> 자식 노드에 의해 왼쪽으로 치우침	LL 회전 : 문제 구간 중 상위 구간을 오른쪽으로 회전시킴
RR 유형 : 불균형 발생 노드의 오른쪽 <sup>Right</sup> 자식 노드와 자식의 오른쪽 <sup>Right</sup> 자식 노드에 의해 오른쪽으로 치우침	RR 회전 : 문제 구간 중 상위 구간을 왼쪽으로 회전시킴
LR 유형 : 불균형 발생 노드의 왼쪽 <sup>Left</sup> 자식 노드와 자식의 오른쪽 <sup>Right</sup> 자식 노드에 의해 왼쪽 서브 트리가 치우침	LR 회전 : 문제 구간 중 하위 구간을 왼쪽으로 1차 회전시켜 LL 유형으로 변환한 다음 LL 회전을 적용함
RL 유형 : 불균형 발생 노드의 오른쪽 <sup>Right</sup> 자식 노드와 자식의 왼쪽 <sup>Left</sup> 자식 노드에 의해 오른쪽 서브 트리가 치우침	RL 회전 : 문제 구간 중 하위 구간을 오른쪽으로 1차 회전시켜 RR 유형으로 변환한 다음 RR 회전을 적용함

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 1 AVL 트리의 회전 연산

### ▶ LL 회전 연산

- 삽입, 삭제 연산 후에 AVL 트리에 LL 유형의 불균형이 발생했을 때 적용



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 1 AVL 트리의 회전 연산

### ▶ RR 회전 연산

- 삽입, 삭제 연산 후에 AVL 트리에 RR 유형의 불균형이 발생했을 때 적용



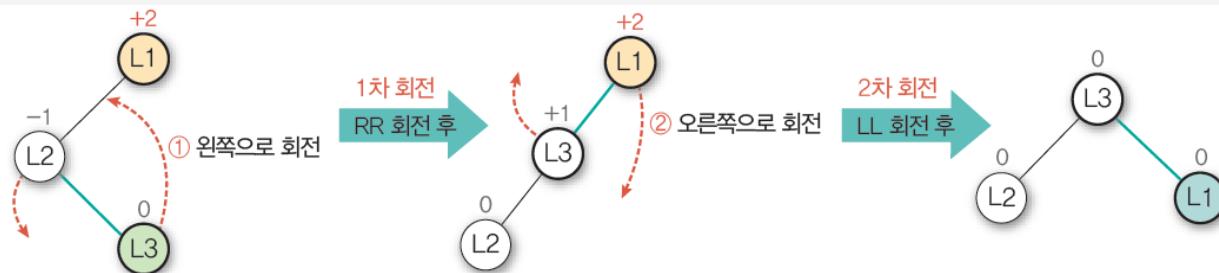
[RR 유형과 회전 예]

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 1 AVL 트리의 회전 연산

### ▶ LR 회전 연산

- 삽입, 삭제 연산 후에 AVL 트리에 LR 유형의 불균형이 발생했을 때 적용



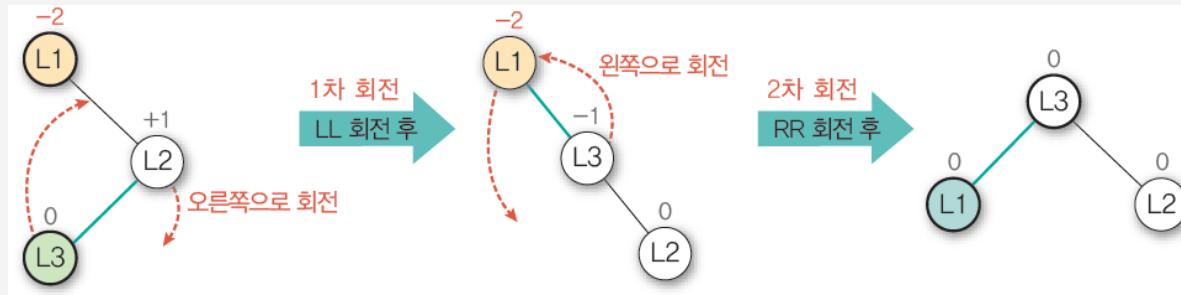
[LR 유형과 LF 회전 예]

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 1 AVL 트리의 회전 연산

### ▶ RL 회전 연산

- 삽입, 삭제 연산 후에 AVL 트리에 RL 유형의 불균형이 발생했을 때 적용



[RL 유형과 회전 예]

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

#### 1 AVL 트리의 회전 연산

##### ▶ AVL 트리 회전 연산 예

- {50, 60, 70, 90, 80, 75, 73, 72, 78}  
원소 아홉 개를 AVL 트리에 삽입하면서 불균형이 발생했을 때 회전 연산을 수행하여 균형 상태를 유지하는 예
  - ✓ (초록색 노드는 삽입 노드, 노란색 노드는 불균형 발생한 노드. 불균형 유형에 해당하는 세 개의 노드는 초록색 간선으로 표시)

#### 1 AVL 트리의 회전 연산

##### ▶ AVL 트리 회전 연산 예

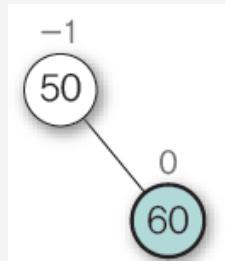
① 노드 50 삽입

: 공백 AVL 트리에 노드 50을 삽입



② 노드 60 삽입

: 노드 50의 오른쪽 자식으로  
노드 60을 삽입



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 1 AVL 트리의 회전 연산

### ▶ AVL 트리 회전 연산 예

- ③ 노드 70 삽입 : 노드 60의 오른쪽 자식으로 노드 70을 삽입,  
하지만 노드 50의 균형 인수가 -2가 되어 RR 유형의 불균형이 발생하므로  
RR 회전을 수행하여 균형을 회복



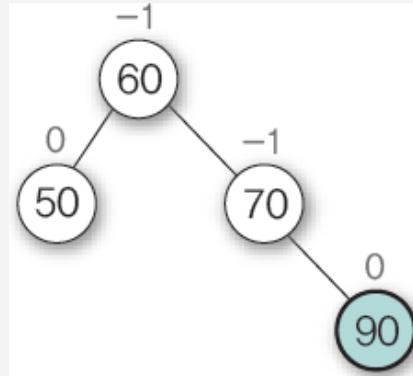
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

#### 1 AVL 트리의 회전 연산

##### ▶ AVL 트리 회전 연산 예

④ 노드 90 삽입

: 노드 70의 오른쪽 자식으로 노드 90을 삽입

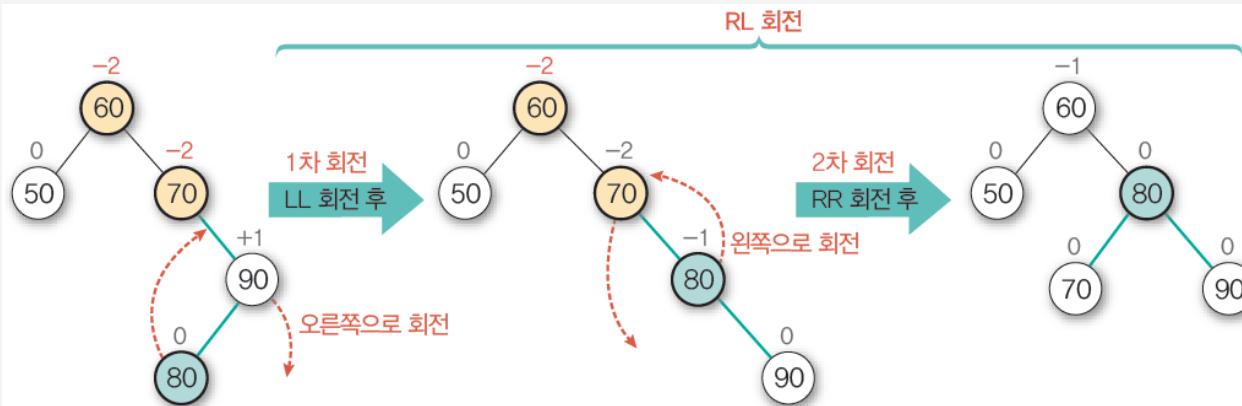


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 1 AVL 트리의 회전 연산

### ▶ AVL 트리 회전 연산 예

- ⑤ 노드 80 삽입 : 노드 90의 왼쪽 자식으로 노드 80을 삽입 후 노드 70의 균형 인수가 -2가 되어 RL유형의 불균형이 발생하므로 1차 회전에서 LL 회전을 하고 2차 회전에서 RR 회전을 하는 RL 회전을 수행하여 균형을 회복

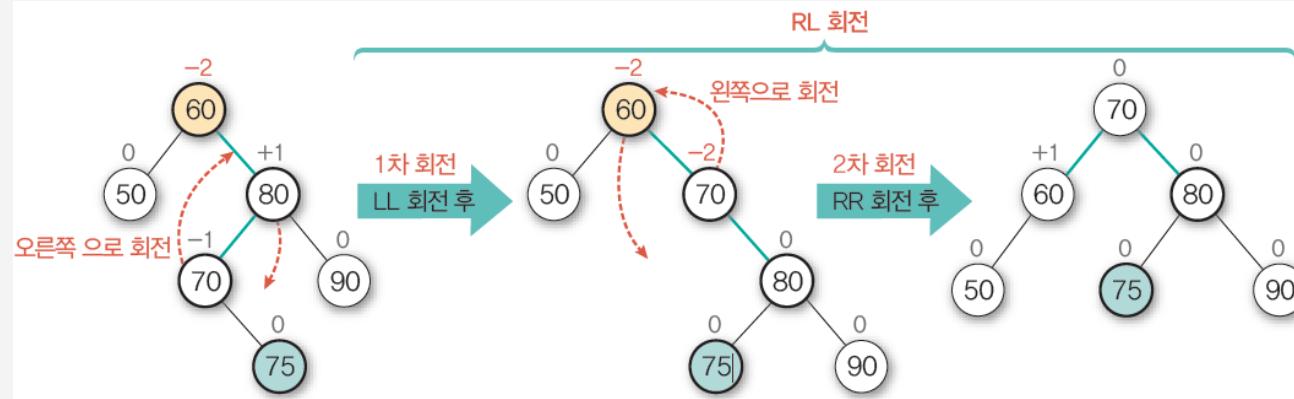


※출처: 이지영(2016). IT CookBook,  
C로 배우는 쉬운 자료구조(개정3판).  
한빛미디어

## 1 AVL 트리의 회전 연산

### ▶ AVL 트리 회전 연산 예

- ⑥ 노드 75 삽입 : 노드 70의 오른쪽 자식으로 노드 75를 삽입 후 노드 60의 균형 인수가 -2가 되었고, RL 유형의 불균형이 발생하였으므로 1차 회전에서 LL 회전을 하고 2차 회전에서 RR 회전을 하는 RL회전을 수행하여 균형을 회복



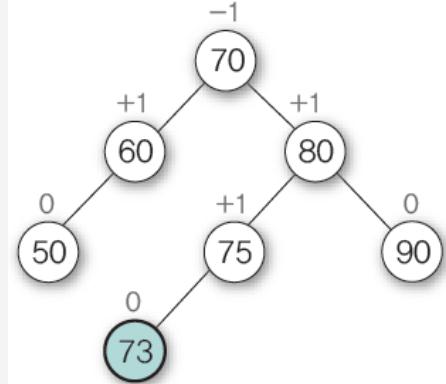
※출처: 이지영(2016). IT CookBook,  
C로 배우는 쉬운 자료구조(개정3판).  
한빛미디어

## 1 AVL 트리의 회전 연산

### ▶ AVL 트리 회전 연산 예

⑦ 노드 73 삽입

: 노드 75의 왼쪽 자식으로 노드 73을 삽입

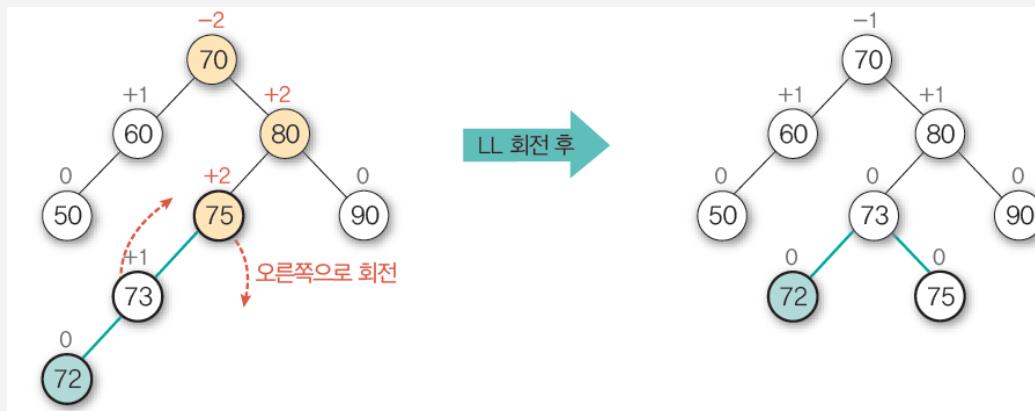


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

## 1 AVL 트리의 회전 연산

### ▶ AVL 트리 회전 연산 예

- ⑧ 72 삽입 : 노드 75의 균형 인수가 +2가 되면서 LL 유형의 불균형이 발생  
LL 회전 연산을 수행하여 균형을 회복



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

### 3 | AVL 트리의 회전 연산

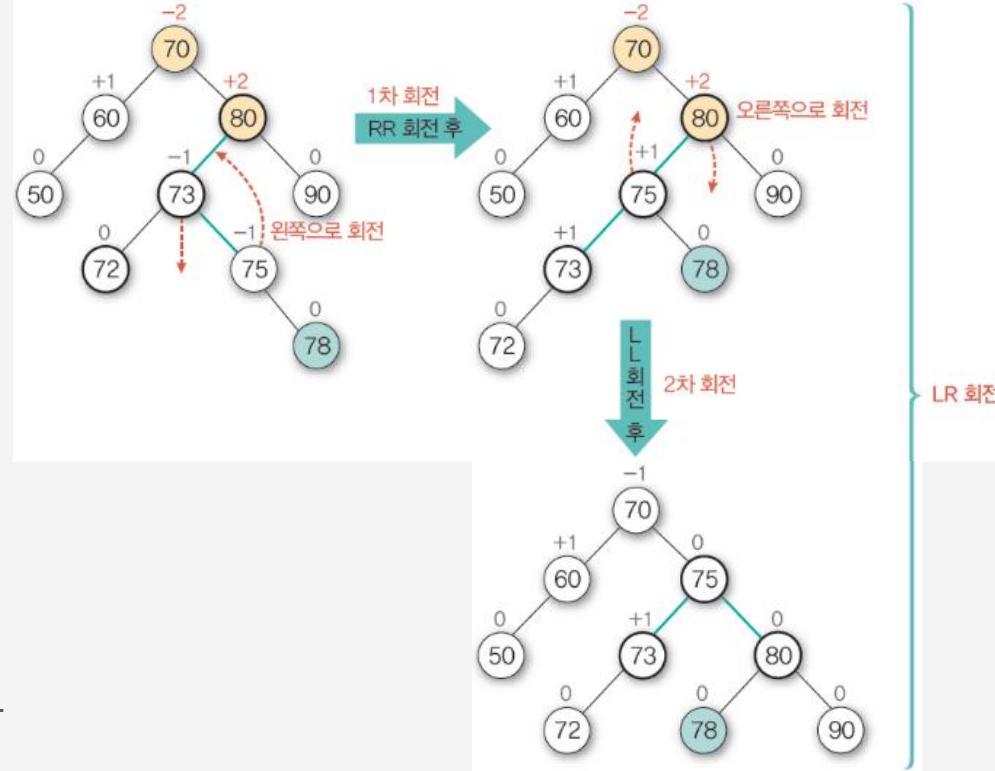
#### 1 AVL 트리의 회전 연산

##### ▶ AVL 트리 회전 연산 예

###### ⑨ 노드 78 삽입 :

노드 75의 오른쪽 자식 노드로 노드 78을 삽입  
노드 80의 균형 인수가 +2가 되면서

LR 유형의 불균형이 발생  
1차 회전에서 RR 회전을  
하고 2차 회전에서  
LL 회전을 하는 LR 회전을  
수행하여 균형을 회복

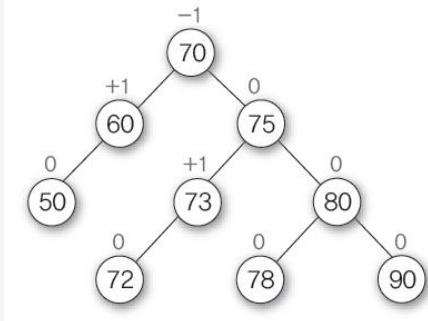


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

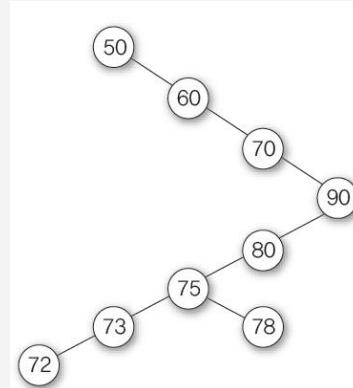


## 1 AVL 트리의 회전 연산

- ▶ 원소 {50, 60, 70, 90, 80, 75, 73, 72, 78} 를 같은 순서로 삽입한 경우의 AVL 트리와 이진 탐색 트리 비교



(a) AVL트리



(b) 이진 트리 탐색

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어