

Data Engineering

# 데이터엔지니어링

[ 트리 기반 데이터 엔지니어링 ]  
- TreeSet CRUD -



# TreeSet 소개 (1)

## - Tree와 BST

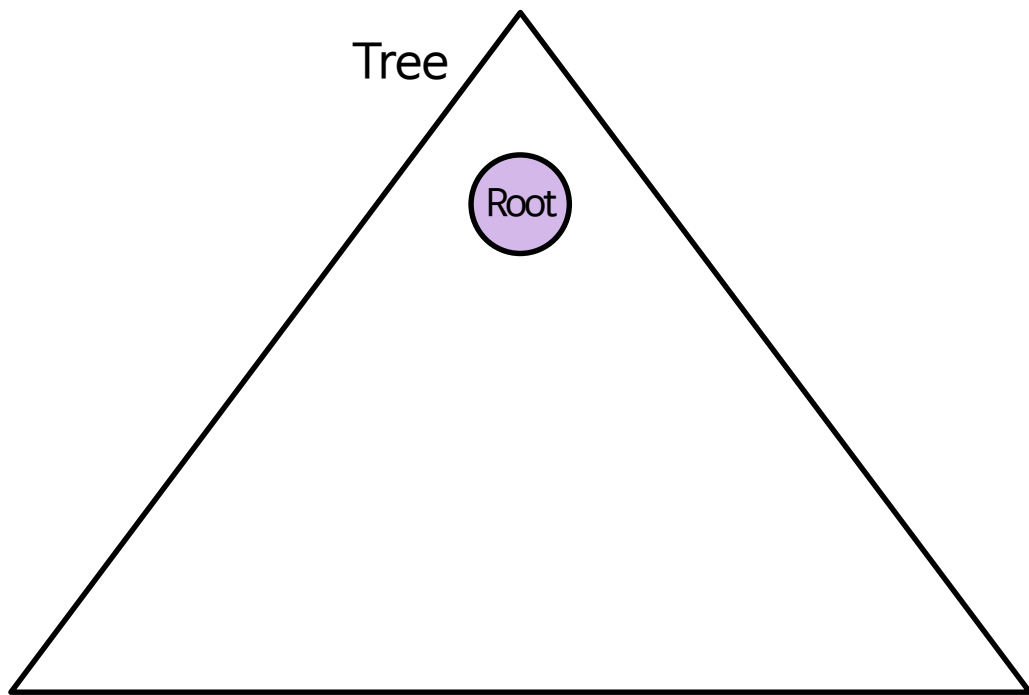
## 학습내용

- 1 TreeSet의 개념
- 2 TreeSet을 구성하는 인터페이스

## 학습목표

- TreeSet의 개념을 설명할 수 있다.
- TreeSet을 구성하는 인터페이스의 연산을 수행할 수 있다.

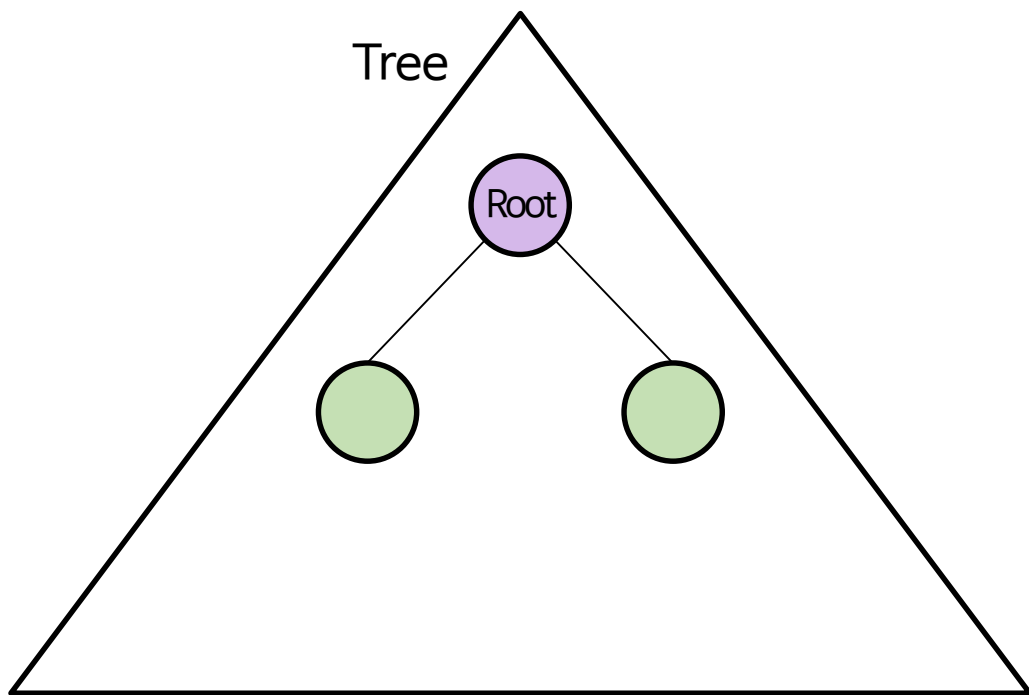
# Tree



Tree는 Node들로 이루어짐

Tree가 비어 있지 않다면,  
Tree는 1개의 Root Node를 가짐

# Tree

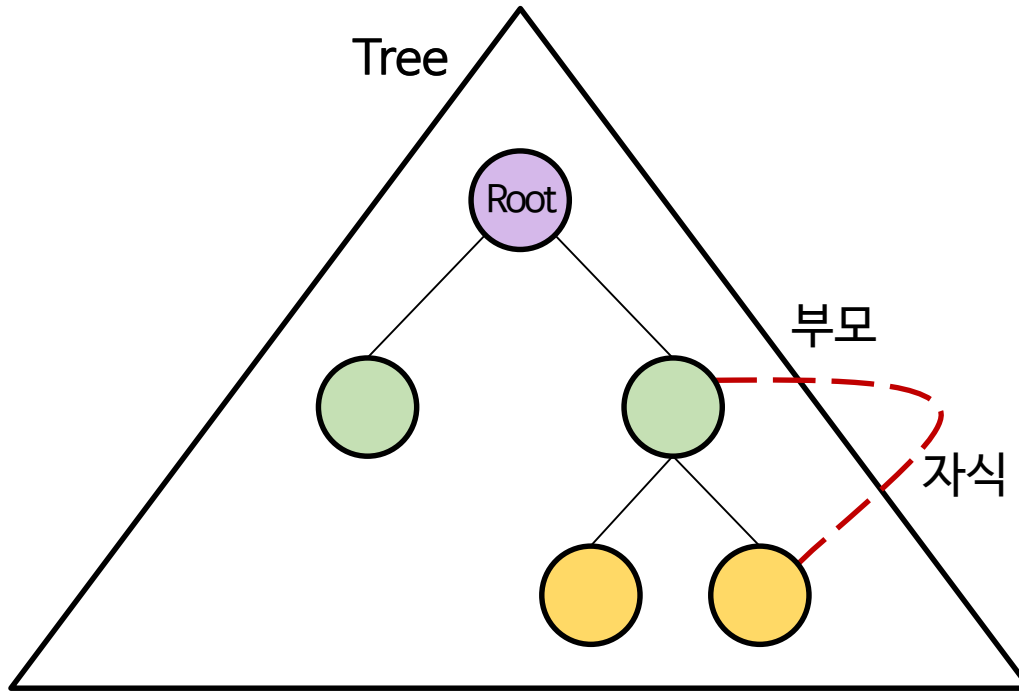


Tree는 Node들로 이루어짐

Tree가 비어 있지 않다면,  
Tree는 1개의 Root Node를 가짐

Root는 부모로서  
Child Node들을 가질 수 있음

# Tree



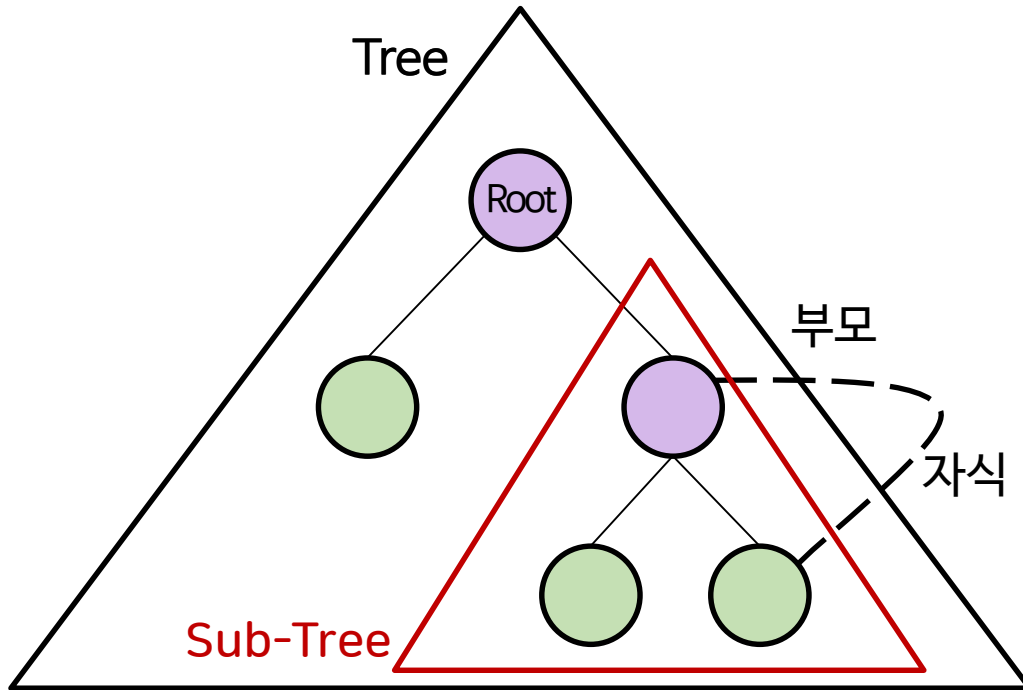
Tree는 Node들로 이루어짐

Tree가 비어 있지 않다면,  
Tree는 1개의 Root Node를 가짐

Root는 부모로서  
Child Node들을 가질 수 있음

Child는 부모로서 Children을 갖는  
Sub-Tree를 형성할 수 있음

# Tree



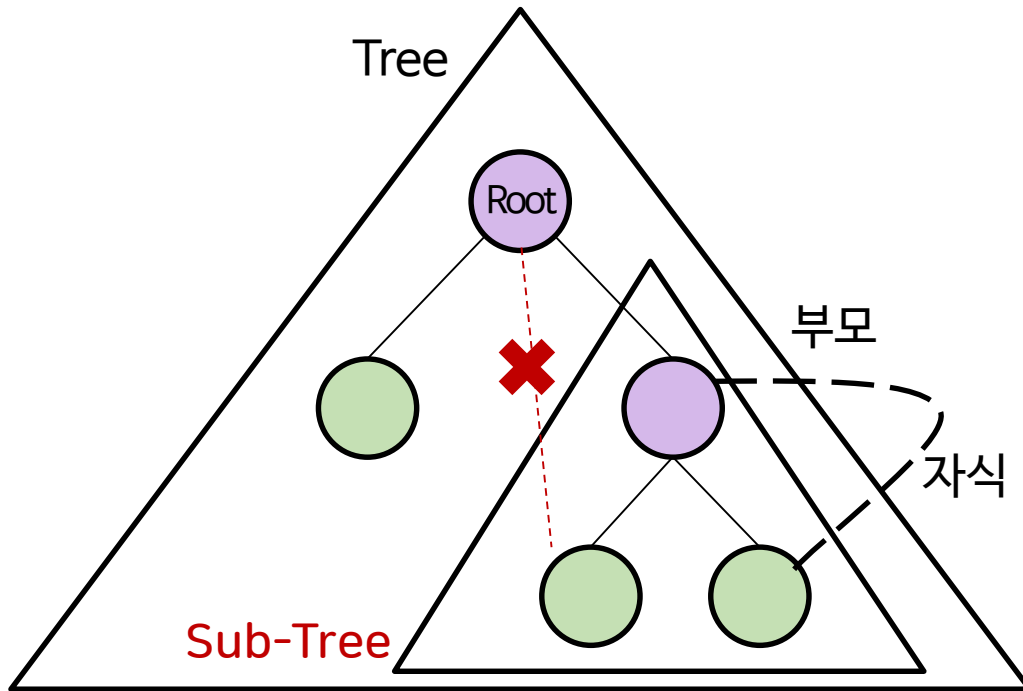
Tree는 Node들로 이루어짐

Tree가 비어 있지 않다면,  
Tree는 1개의 Root Node를 가짐

Root는 부모로서  
Child Node들을 가질 수 있음

Child는 부모로서 Children을 갖는  
Sub-Tree를 형성할 수 있음

# Tree



Tree는 Node들로 이루어짐

Tree가 비어 있지 않다면,  
Tree는 1개의 Root Node를 가짐

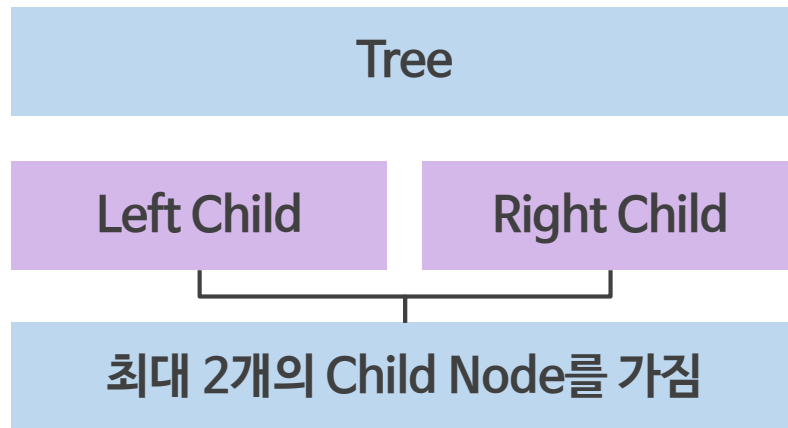
Root는 부모로서  
Child Node들을 가질 수 있음

Child는 부모로서 Children을 갖는  
Sub-Tree를 형성할 수 있음

Root는 부모가 없으며,  
그 외의 노드는 부모가 하나임

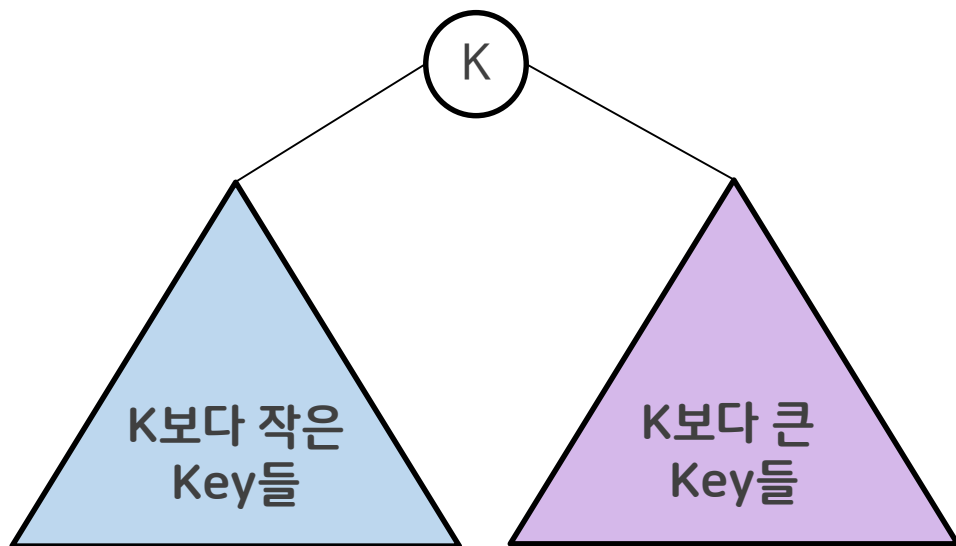


# Binary Search Tree(BST)



# Binary Search Tree(BST)

## Key를 가짐



특정 Node의 Left Sub-Tree의 Key

➡ Node의 Key보다 전부 작음

특정 Node의 Right Sub-Tree의 Key

➡ Node의 Key보다 전부 큼

# Binary Search Tree(BST)

최솟값 :  $O(h)$

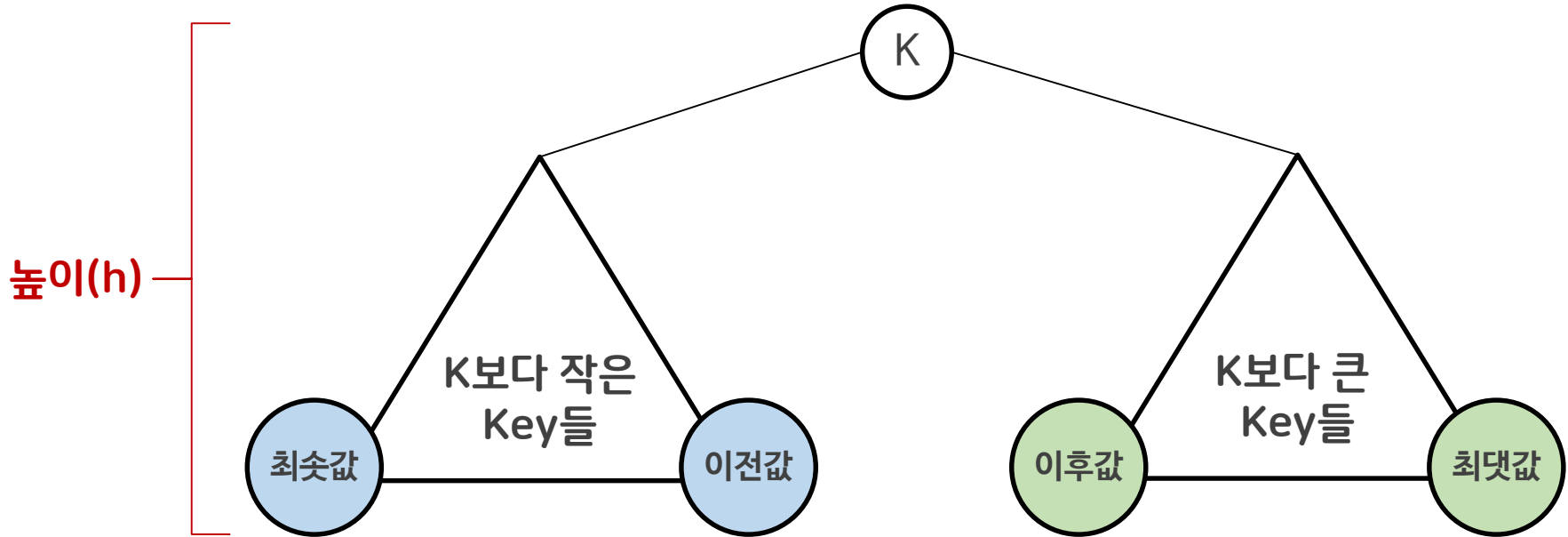
최댓값 :  $O(h)$

이전값 :  $O(h)$

이후값 :  $O(h)$

오름차순/내림차순 순회 :  $O(n)$

# Binary Search Tree(BST)



# Binary Search Tree(BST)

Tree가 한쪽으로 치우칠 경우



높이( $O(h)$ ) = 자료의 개수 ( $O(n)$ )

**최대값, 최소값 등의 문제도  
자료의 개수에 비례하여 성능 결정**

# Binary Search Tree(BST)

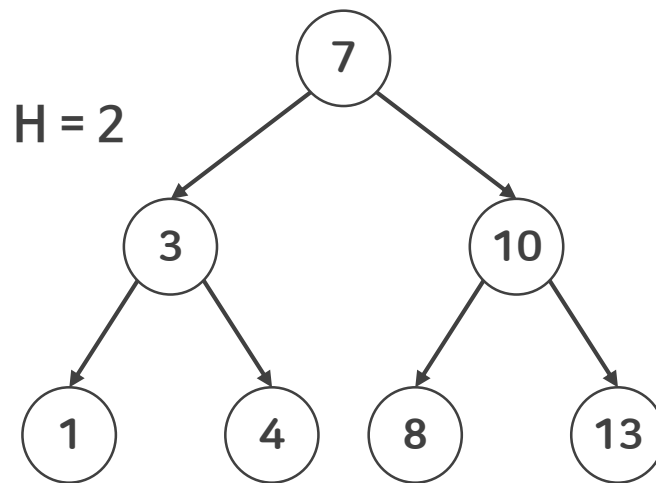
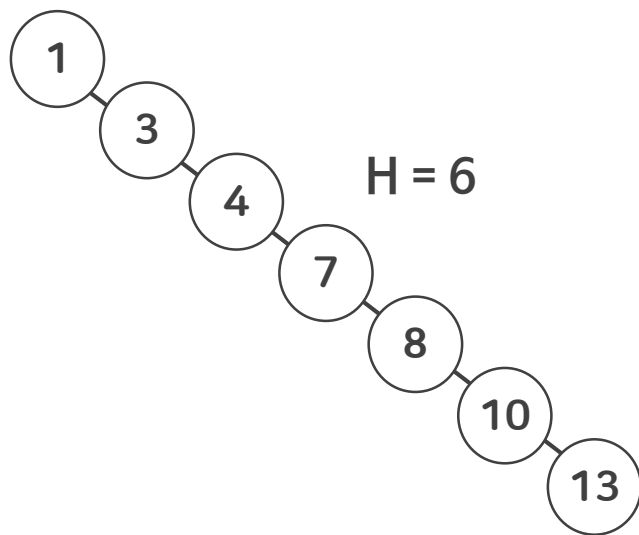
높이를 낮추기 위한 (Balanced Tree) 방법

AVL

Red-black Tree

B-Tree

B+-Tree



# Remind

---

Tree

Binary Search Tree(BST)