

# **자료구조 실습 보고서**

[제 12 주]

제출일 : 2018.5.30.수요일

학번 : 201604137

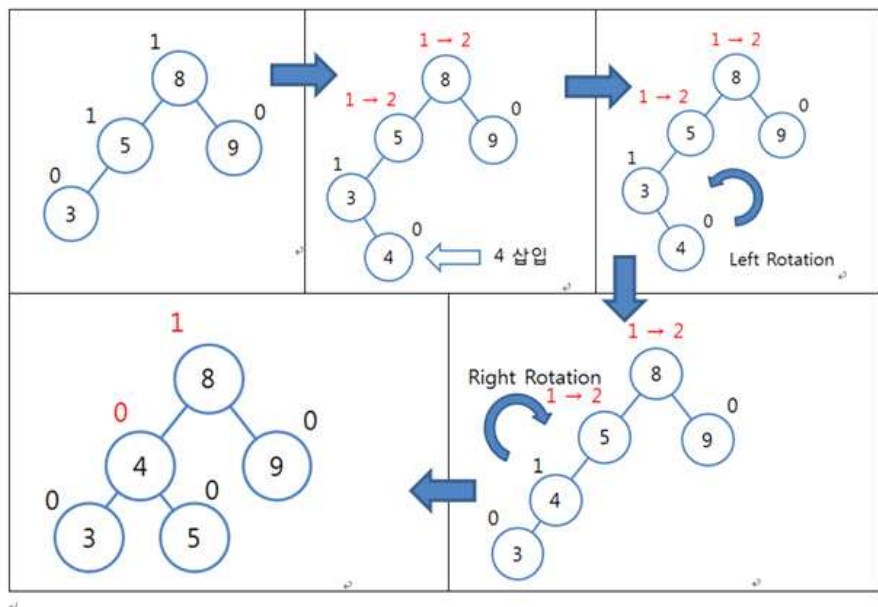
이름 : 김예지

## [프로그램 설명서]

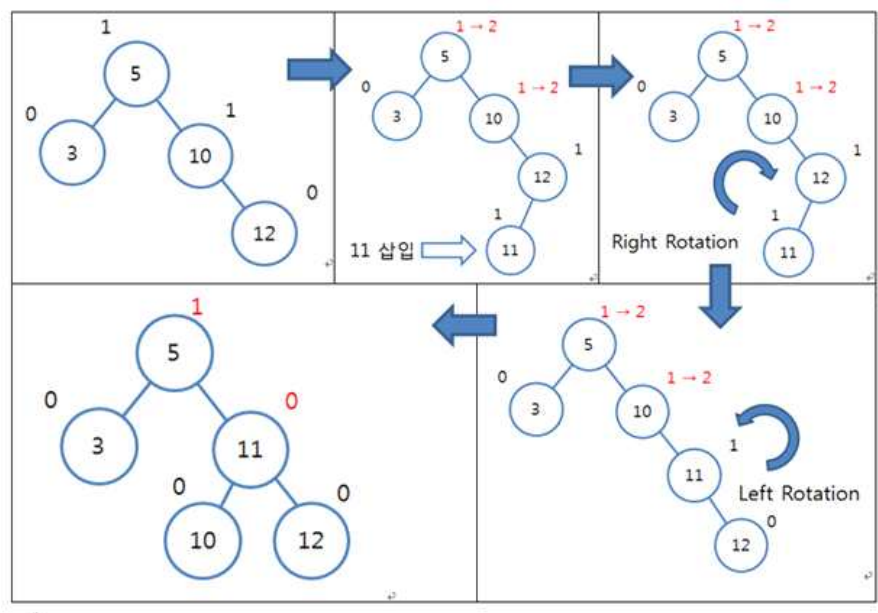
### 이진 검색 트리(binary search tree)

- ① 왼쪽 자식이 있다면, 왼쪽 자식의 key 값보다 자신의 key 값이 커야 한다.
- ② 오른쪽 자식이 있다면, 오른쪽 자식의 key 값보다 자신의 key 값이 작아야 한다.

#### 〈LR회전〉



#### 〈RL회전〉

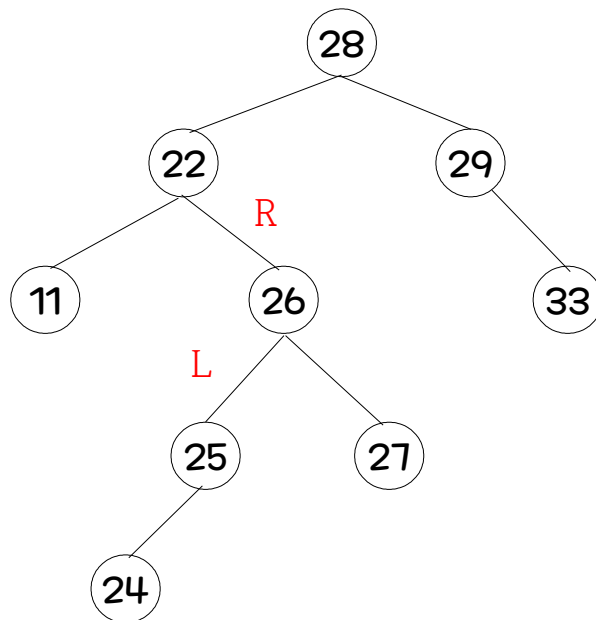
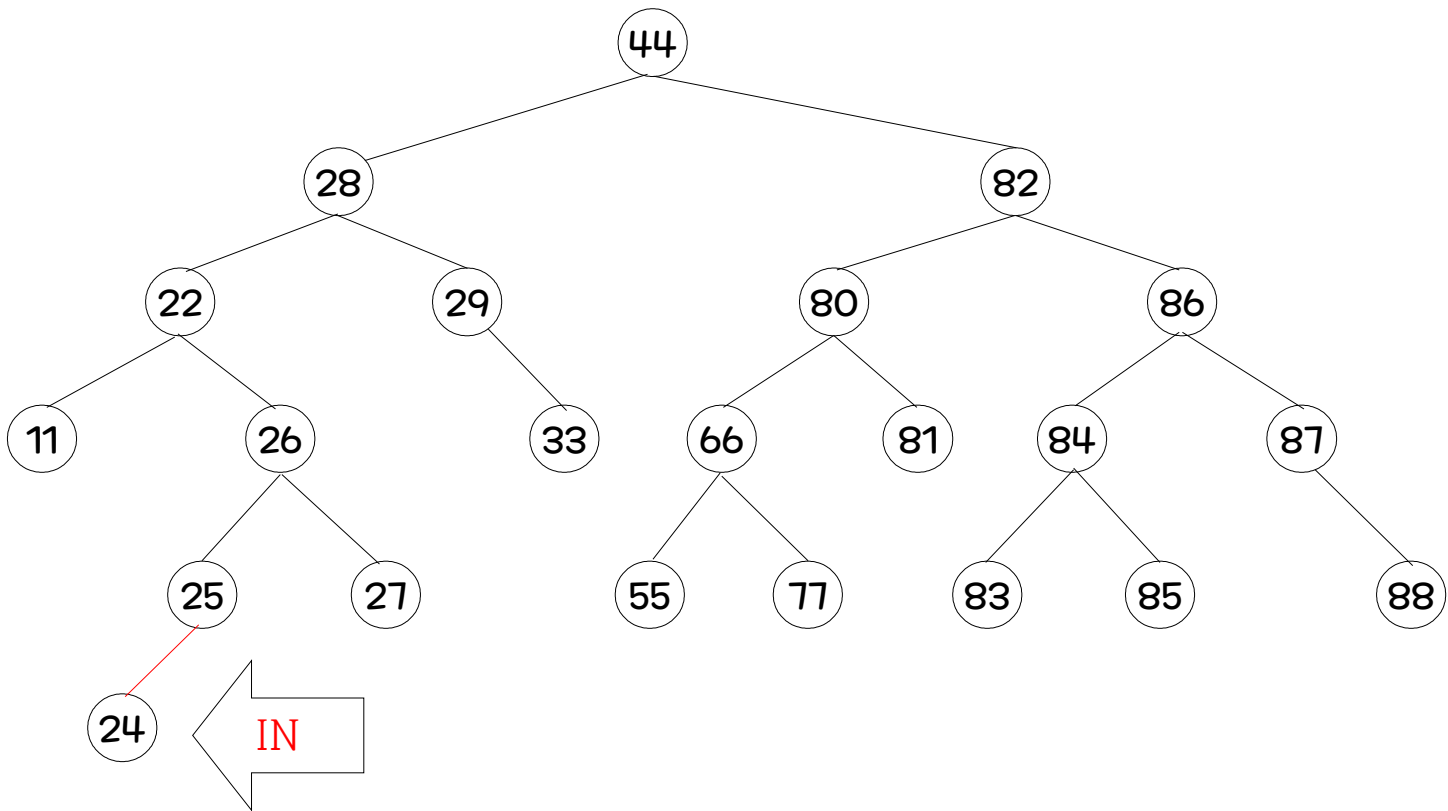


## [실행 결과 분석]-(1) BST Tree

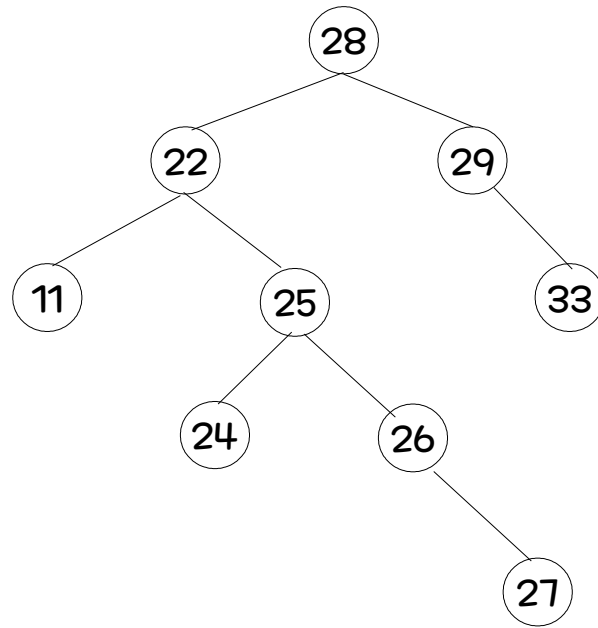
```
public Boolean isBST() {  
  
    if(!(root instanceof Comparable)) return false;  
    Comparable key = (Comparable) root;  
  
    //key값이 왼쪽트리들의 루트보다 작으면 안됨  
    if(key.compareTo(this.left.getRoot())<0) return false;  
  
    return true;  
}
```

```
public void insertKey(Object data) {  
    Comparable key = (Comparable) root;  
    if(key.compareTo(data)<0) { //추가하려는 데이터가 root보다 클 경우 (오른쪽)  
        //root의 right가 null이 아닐 때 재귀호출  
        if(this.right != null) {  
            this.right.insertKey(data);  
        }  
        //root의 right가 null일 때 객체 생성 후 값을 넣는다.  
        else {  
            right = new BinaryTree(data);  
        }  
    }  
    else { //추가하려는 데이터가 root보다 작을 경우 (왼쪽)  
        //root의 left가 null이 아닐 때  
        if(this.left != null) {  
            this.left.insertKey(data);  
        }  
        //root의 left가 null일 때  
        else {  
            left = new BinaryTree(data);  
        }  
    }  
}
```

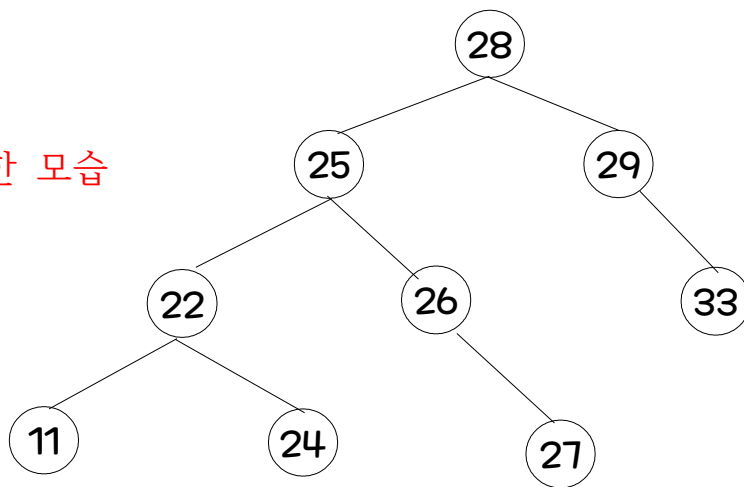
[실행 결과 분석]-(2)AVL Tree 24 삽입 과정



R회전 한 모습



L회전 한 모습



최종

