

자료구조

11

과목명 : 자료구조

분반 : 06

담당교수 : 박정희

학번 : 202002546

이름 : 임우진

목차

- 1. 문제 요구 사항**
- 2. 풀이**
- 3. 실행 화면 캡처**

1. 문제 요구 사항

입력 파일의 문자열을 받아 AVL트리에 삽입하고

k번째 key를 삭제하는 메소드를 구현하고

왼쪽 서브 트리에 있는 원소의 개수를 출력하는 메소드도 구현해서
출력 양식에 맞게 출력한다.

순회 방식은 전위 순회(preorder)를 사용한다.

1) Node.java 구현

2) AVLTree.java 구현 kdelete, getLeftsize 메소드 구현

3) MainJava.java 작성

2. 풀이

Node.java

필드

타입	필드 이름
Key	id
Value	name
Int	height
Node<Key,Value>	left
Node< Key,Value>	right
Int	leftsize

메소드

```
public int height(Node n) { //높이
    return n.height;
}
public int getHeight(){ //높이
    return height;
}

public Node(Key newID, Value newName, int newHt){ //생성자
    id = newID;
    name = newName;
    height = newHt;
    left = right = null;
    leftsize = 0;
}

public Key getKey(){
    return id;
}
```

```
public Node<Key,Value> getLeft(){ return left; }

public Node<Key,Value> getRight(){ return right; }

public void setLeft(Node<Key,Value> lt){
    left = lt;
    updateLeftSize();
}

public void setRight(Node<Key,Value> rt){
    right = rt;
    updateLeftSize();
}

private void updateLeftSize() {
    int leftSize = (left != null) ? left.getLeftSize() + 1 : 0;
    leftsize = leftSize;
}

public void setHeight(int i) {
    this.height = i;
}

public int getLeftSize() {
    return left != null ? left.leftsize + 1 : 0;
}
```

AVLTree.java

주요 메소드

1) rotateRight

```
private Node<Key,Value> rotateRight(Node<Key,Value> n){
    Node<Key,Value> x = n.getLeft();
    n.setLeft(x.getRight());
    x.setRight(n);
    n.setHeight(tallerHeight(height(n.getLeft()),
height(n.getRight())) + 1);
    x.setHeight(tallerHeight(height(x.getLeft()),
height(n.getRight())) + 1);
    return x;
}
```

2) rotateLeft

```
private Node<Key,Value> rotateLeft(Node<Key,Value> n){
    Node<Key, Value> x = n.right;
    n.setRight(x.getLeft());
    x.setLeft(n);
    n.setHeight(tallerHeight(height(n.getLeft()),
height(n.getRight())) + 1);
    x.setHeight(tallerHeight(height(x.getLeft()),
height(x.getRight())) + 1);
    return x;
}
```

3) getLeftSize

```
public int getLeftSize() {
    if (root == null) {
        return 0;
    }
    return size(root.getLeft());
}
```

4) deleteKth

```
public void deleteKth(int k) {  
    if (root == null) {  
        System.out.println("empty tree");  
        return;  
    }  
}
```

```
private Node<Key, Value> deleteKth(Node<Key, Value> node, int k) {  
    if (node == null) {  
        return null;  
    }  
  
    int leftSize = size(node.getLeft()) + 1;  
  
    if (k < leftSize) {  
        node.setLeft(deleteKth(node.getLeft(), k));  
        node.leftsize--;  
    } else if (k > leftSize) {  
        node.setRight(deleteKth(node.getRight(), k - leftSize));  
    } else {  
        if (node.getLeft() == null && node.getRight() == null) {  
            return null;  
        } else if (node.getLeft() == null) {  
            return node.getRight();  
        } else if (node.getRight() == null) {  
            return node.getLeft();  
        } else {  
            Node<Key, Value> N_Node = min(node.getRight());  
            N_Node.setRight(deleteMin(node.getRight()));  
            N_Node.setLeft(node.getLeft());  
            node = N_Node;  
        }  
    }  
}  
  
node.setHeight(tallerHeight(height(node.getLeft()),  
height(node.getRight())) + 1);  
return balance(node);  
}
```

3. 실행 화면 캡처

```
C:\Users\8146q\.jdk\openjdk-19.0.2\bin\java.exe
4 2 1 3 8 6 5 7 9 10
3
=====
4 2 3 8 6 5 7 9 10
2

=====
7 3 2 1 5 4 6 9 8 10
6
=====
7 3 2 1 5 4 6 9 10
6

=====

종료 코드 0(으)로 완료된 프로세스
```