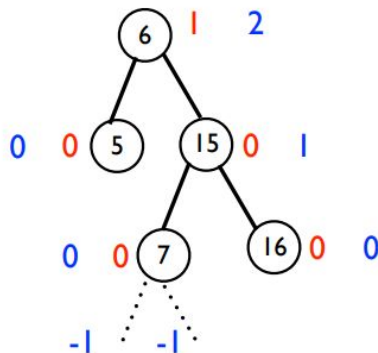# Lab 06: AVL Tree

Data Structure 2023

AVL Tree

- Binary search tree

- For every node in the tree, the **heights** of its left subtree and right subtree differ by **at most 1**.
  - the height of a null subtree is –1
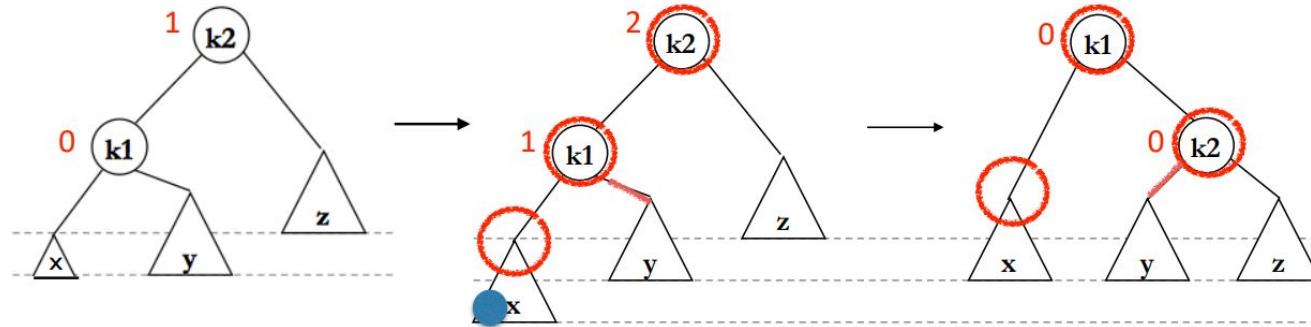 - the height of a subtree with one node is 0

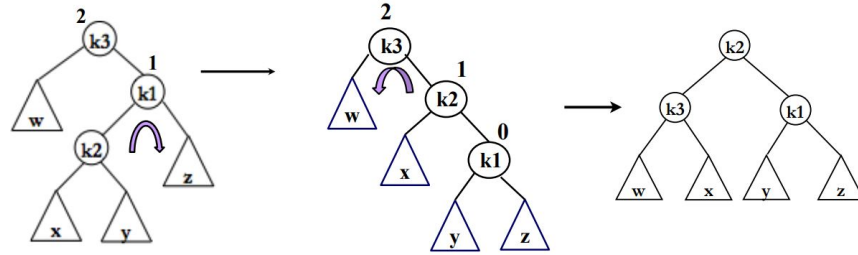
h = height of the subtree
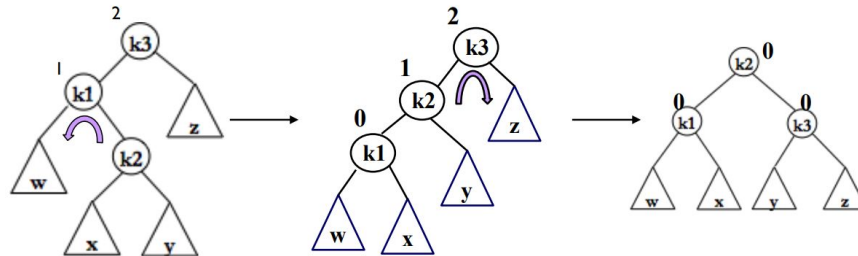
$diff = |h_L - h_R|$

AVL Tree

- Single Rotation

AVL Tree

- Double Rotation

right-left double notation

Left-right double notation

AVL Tree ADT

- **Insert** Insert a new node with the key value into the AVL tree. If the key already exists in the tree, print an error message.

- **Delete** Delete a node with the given key value from the AVL tree. Reform tree using right subtree. If the key does not exist in the tree, print an error message.

- **PrintPreorder** Print the tree by preorder traversal.

- **DeleteTree** Delete the tree.

AVL Tree ADT

Structure

```
struct AVLNode;

typedef struct AVLNode *Position;

typedef struct AVLNode *AVLTree;

typedef int ElementType;


typedef struct AVLNode{

    ElementType element;

    AVLTree left, right;

    int height;

}AVLNode;
```

Function

```
AVLTree Insert(ElementType X, AVLTree T);

AVLTree Delete(ElementType X, AVLTree T);

Position SingleRotateWithLeft(Position node);

Position SingleRotateWithRight(Position node);

Position DoubleRotateWithLeft(Position node);

Position DoubleRotateWithRight(Position node);

void PrintPreorder(AVLTree T);

void DeleteTree(AVLTree T);
```

Input & Output Example


```
input1.txt
lab07_AVL_edit > input1.txt
 1   i 1
 2   i 2
 3   i 2
 4   i 3
 5   d 3
 6   i 8
 7   i 5
 8   i 9
 9   d 5
10   i 22
11   d 4
12   i 11
13
```

```
output1.txt
lab07_AVL_edit > output1.txt
 1   1(0)
 2   1(1) 2(0)
 3   insertion error : 2 is already in the tree!
 4   1(1) 2(0)
 5   2(1) 1(0) 3(0)
 6   2(1) 1(0)
 7   2(1) 1(0) 8(0)
 8   2(2) 1(0) 8(1) 5(0)
 9   2(2) 1(0) 8(1) 5(0) 9(0)
10   2(2) 1(0) 8(1) 9(0)
11   2(2) 1(0) 9(1) 8(0) 22(0)
12   deletion error : 4 is not in the tree!
13   2(2) 1(0) 9(1) 8(0) 22(0)
14   9(2) 2(1) 1(0) 8(0) 22(1) 11(0)
15
```

HANYANG UNIVERSITY

Assignment

- Due
  - **~ 2023.04.19(Wed) 23:59**
  - Last Commit 기준

- 자세한 내용은 과제 명세 PDF 파일 참고