Lab7

Lab7

- The deadline for lab7 submission is April 29 at 11:59 pm.
- If you have any question, please contact TA.

- Folder name : lab7
- Code name: p7.c
- Each code will be tested by 5 different input files.
- 20 score for each input. If you don't get the answer, you get 0 score.

Evaluation criteria

Category	Evaluation	
p7	100	
Total	100	

- *Use GCC **11** version.*
- *No score will be given if the gcc version is different.*

Lab7. AVL Tree

AVLTree Insert(ElementType X, AVLTree T) insert a new node to the AVL Tree.

void PrintInorder(AVLTree T) print the tree by in-order traversal. Print height of the node inside bracket.

void DeleteTree(AVLTree T) free tree. (code will be provided)

Position SingleRotateWithLeft(Position node)

Position SingleRotateWithRight(Position node)

Position DoubleRotateWithLeft(Position node)

Position DoubleRotateWithRight(Position node)

Lab7. AVL Tree

Exception errors that need to be printed.

- Insert
  - key duplication – “Insertion Error: [key] already in the tree!”

Lab7. AVL Tree

- Structure

```
#include <stdio.h>
#include <stdlib.h>

struct AVLNode;
typedef struct AVLNode *Position;
typedef struct AVLNode *AVLTree;
typedef int ElementType;

struct AVLNode{
        ElementType Element;
        AVLTree Left;
        AVLTree Right;
        int Height;
};
```

- Function

```
int Max(ElementType num1, ElementType num2);
int Height(Position P);
Position SingleRotateWithLeft(Position node);
Position SingleRotateWithRight(Position node);
Position DoubleRotateWithLeft(Position node);
Position DoubleRotateWithRight(Position node);
AVLTree Insert(ElementType X, AVLTree T);
void PrintInorder(AVLTree T);
void DeleteTree(AVLTree T);
```

Lab7. AVL Tree

```
int main(int argc, char **argv){

    AVLTree myTree = NULL;
    int key;

    FILE *fi = fopen(argv[1], "r");
    while (fscanf(fi, "%d", &key) != EOF){
        myTree = Insert(key, myTree);
        PrintInorder(myTree);
        printf("₩n");
    }
    fclose(fi);

    DeleteTree(myTree);
    return 0;
}
```

```
void DeleteTree(AVLTree T){
    if (T->Left != NULL)
        DeleteTree(T->Left);
    if (T->Right != NULL)
        DeleteTree(T->Right);
    free(T);
}
```

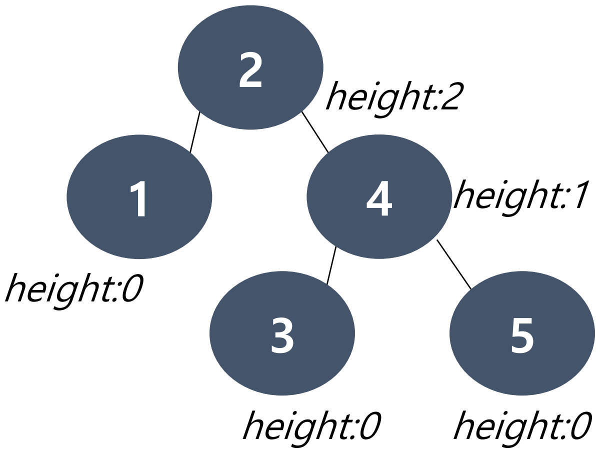
Lab7. AVL Tree

- input : a list of numbers in a file.
- output : the corresponding result in the standard output.

Lab7. AVL Tree – Simple Example

- input file : lab7_input.txt

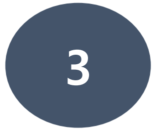
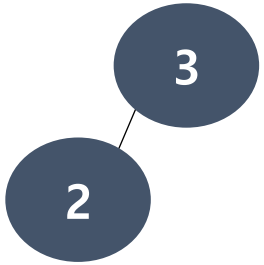
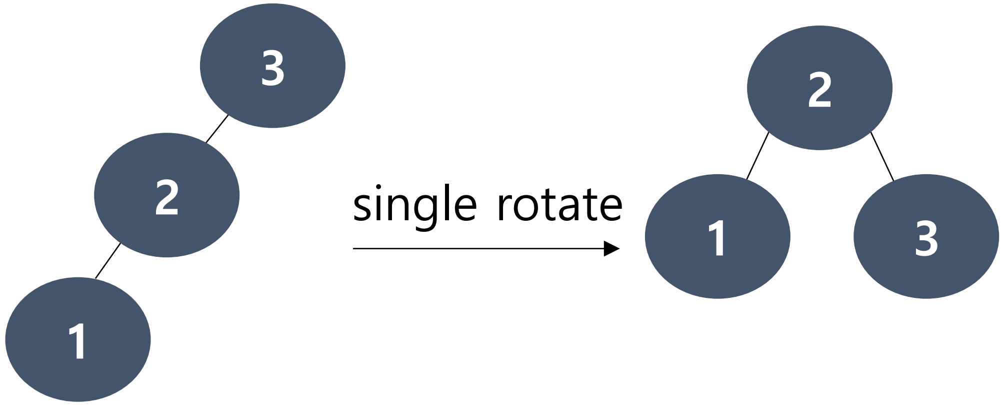
```
3 2 1 4 5
```



- Result

```
3(0)
2(0) 3(1)
1(0) 2(1) 3(0)
1(0) 2(2) 3(1) 4(0)
1(0) 2(2) 3(0) 4(1) 5(0)
```

Lab7. AVL Tree – Simple Example

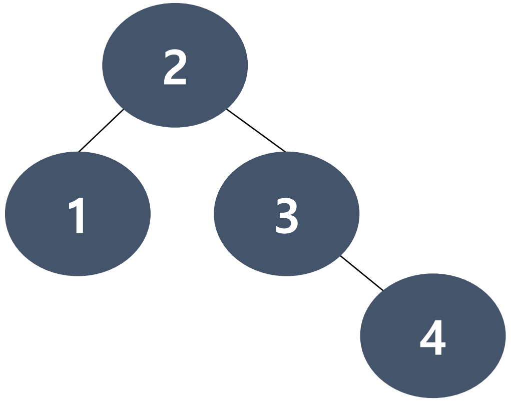
Input		In-order
3		3(0)
3 2		2(0) 3(1)
3 2 1		1(0) 2(1) 3(0)

Lab7. AVL Tree – Simple Example

Input

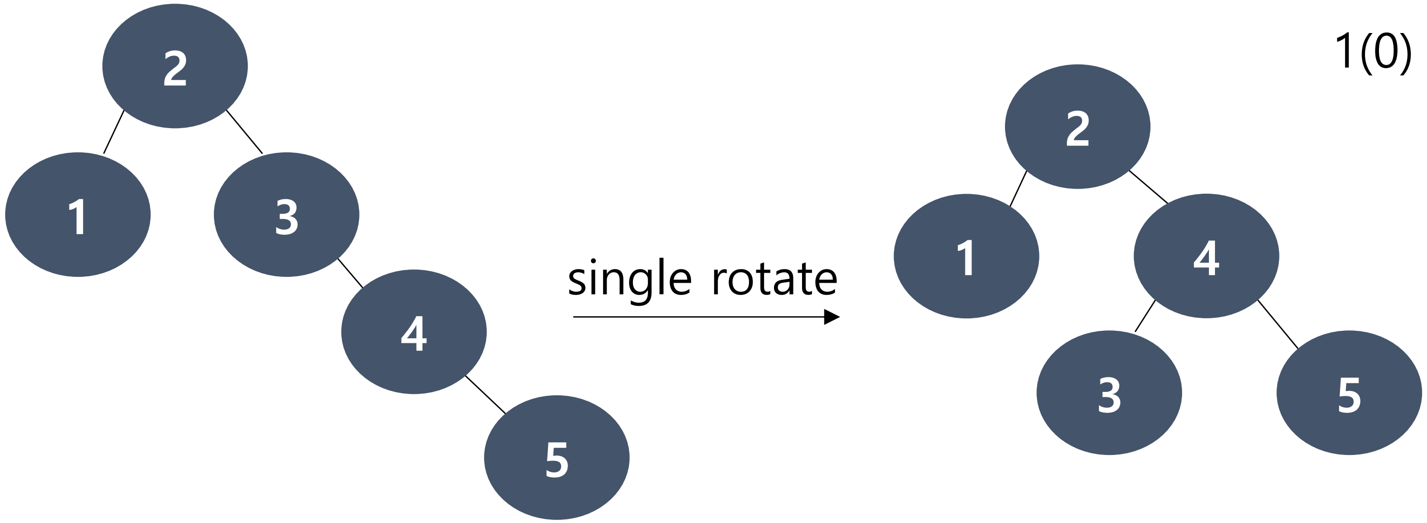
In-order

3 2 1 4



1(0) 2(2) 3(1) 4(0)

3 2 1 4 5



1(0) 2(2) 3(0) 4(1) 5(0)

Lab7. AVL Tree – Example

- input file : lab7_input.txt

```
7 5 3 10 23 4 20 21 22 23 24 25
```

- Result

```
7(0)
5(0) 7(1)
3(0) 5(1) 7(0)
3(0) 5(2) 7(1) 10(0)
3(0) 5(2) 7(0) 10(1) 23(0)
3(1) 4(0) 5(2) 7(0) 10(1) 23(0)
3(1) 4(0) 5(3) 7(0) 10(2) 20(0) 23(1)
3(1) 4(0) 5(3) 7(0) 10(2) 20(0) 21(1) 23(0)
3(1) 4(0) 5(3) 7(0) 10(1) 20(0) 21(2) 22(0) 23(1)
Insertion Error: 23 already in the tree!
3(1) 4(0) 5(3) 7(0) 10(1) 20(0) 21(2) 22(0) 23(1)
3(1) 4(0) 5(3) 7(0) 10(1) 20(0) 21(2) 22(0) 23(1) 24(0)
3(1) 4(0) 5(2) 7(0) 10(1) 20(0) 21(3) 22(0) 23(2) 24(1) 25(0)
```