

Lab 6

AVL Tree

2019. 04. 11

lab 6. AVL Tree

- **Data Structure Specification**

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

- **Function specification**

- int Height(Position P)
- AVLTree Insert(ElementType X, AVLTree T)
- Position SingleRotateWithLeft(Position K)
- Position SingleRotateWithRight(Position K)
- Position DoubleRotateWithLeft(Position K)
- Position DoubleRotateWithRight(Position K)

Binary Search Tree

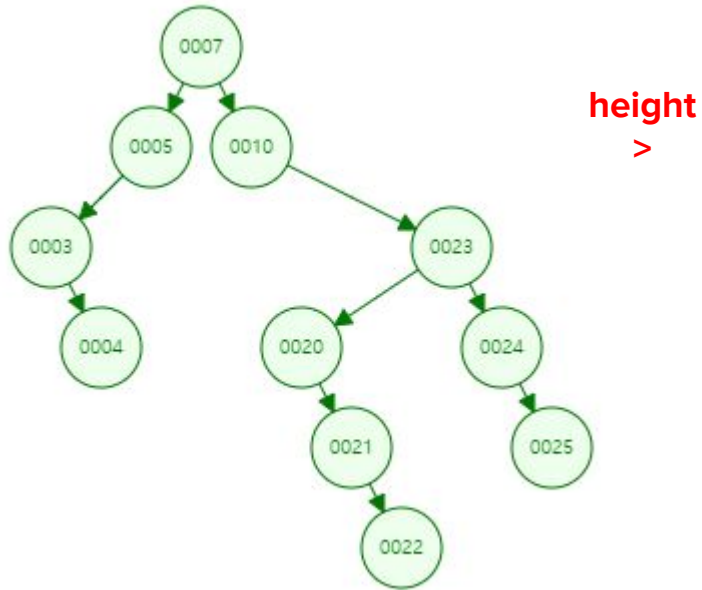
[링크](#)

AVL Tree

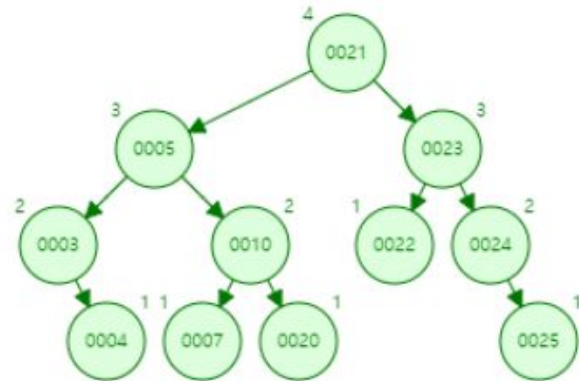
[링크](#)

lab 6. AVL Tree

- Binary Search Tree



- AVL Tree



E.g.

insert 7, 5, 3, 10, 23, 4, 20, 21, 22, 24, 25 in order

lab 6. AVL Tree

- Implement insert function for an AVL tree.
 - Insert
 - Everytime you insert a node, print the tree with height of node by **inorder** traversal
- You have to use file I/O like the previous assignment.

- Input



```
input - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
7 5 3 10 23 4 20 21 22 23 24 25
```

- Output



```
output - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
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(1)
3(1) 4(0) 5(3) 7(0) 10(1) 20(0) 21(2) 22(0) 23(1)
23 already in the tree!
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)
```

lab 6. AVL Tree

- **Input**

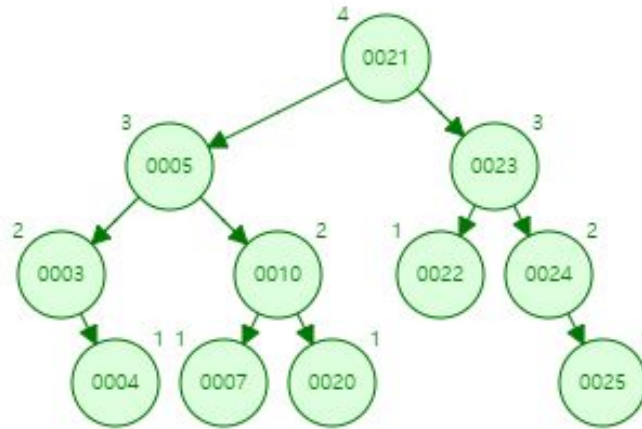
- Obtain a list of elements from the given input file, and insert the given element to tree.
- Each element and the necessary parameters are separated by a space.
- Input values (represented as x below) are any integer.

- **Output**

- You have to print the tree with height of node in parentheses by **inorder** traversal.
- If you try to insert a key value that already exists, print that it already exists.
- ***The leaf node's height should be zero!!***

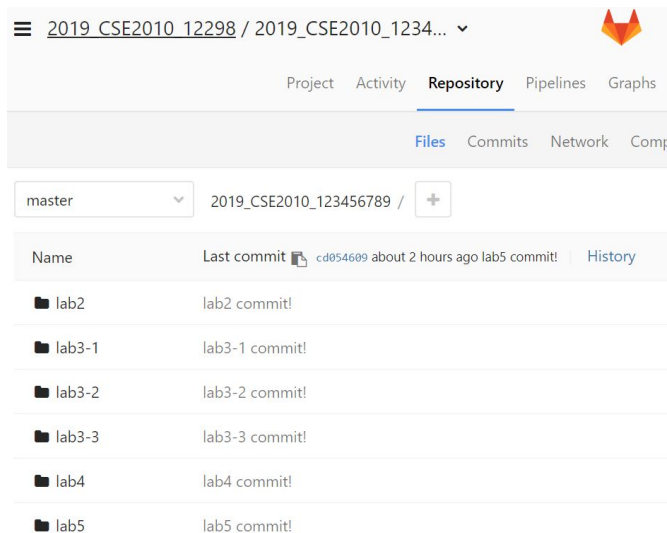
lab 6. AVL Tree

- After inserting the last value, the tree should look like this.



lab 6. AVL Tree

- Submission
 - Project directory name : lab6
 - Source file name : p6.c
 - Executable file name : p6.out
 - You should upload in the hconnect (Gitlab) server.




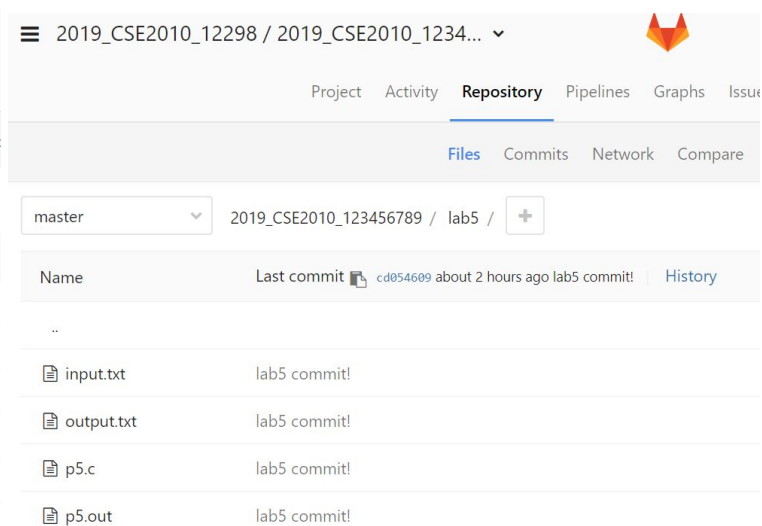
2019_CSE2010_12298 / 2019_CSE2010_1234... ▾

Project Activity **Repository** Pipelines Graphs

Files Commits Network Compare

master ▾ 2019_CSE2010_123456789 / +

Name	Last commit  cd054609 about 2 hours ago lab5 commit! History
lab2	lab2 commit!
lab3-1	lab3-1 commit!
lab3-2	lab3-2 commit!
lab3-3	lab3-3 commit!
lab4	lab4 commit!
lab5	lab5 commit!




2019_CSE2010_12298 / 2019_CSE2010_1234... ▾

Project Activity **Repository** Pipelines Graphs Issues

Files Commits Network Compare

master ▾ 2019_CSE2010_123456789 / lab5 / +

Name	Last commit  cd054609 about 2 hours ago lab5 commit! History
..	
input.txt	lab5 commit!
output.txt	lab5 commit!
p5.c	lab5 commit!
p5.out	lab5 commit!

DeadLine

Wednesday, 17 April, 23 : 59 pm