Lab 5

Binary Search Tree

2019. 04. 04

HYU 한양대학교 HANYANG UNIVERSITY

lab 5. Binary Search Tree

- **Data Structure Specification**

```
typedef struct TreeNode *treeptr;
typedef struct TreeNode{
        int value;
        treeptr *left, *right;
}TreeNode;
```

- **Function specification**
    - Tree* InsertNode(int value, Tree *root)
    - Tree* DeleteNode(int value, Tree *root)
    - Tree* FindNode(int value, Tree *root)
    - void PrintInorder(Tree *root)
    - void PrintPreorder(Tree *root)
    - void PrintPostorder(Tree *root)

# Reference
https://www.cs.usfca.edu/~galles/visualization/BST.html

lab 5. Binary Search Tree

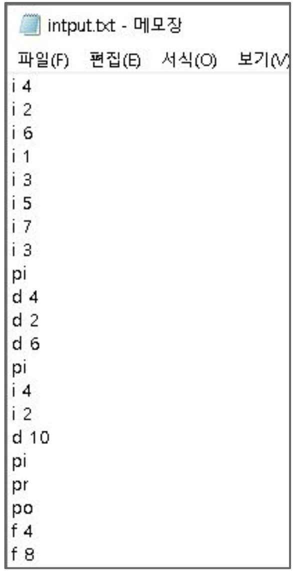
- **Implement binary search tree with the three main functions.**
  - Insert
  - Delete - (using FindMax in the left subtree)
  - Find
  - Additionally, we will have three print functions with different ways of traversal.
    - print the tree by **inorder** traversal
    - print the tree by **preorder** traversal
    - print the tree by **postorder** traversal
- **You have to use file I/O like the previous assignment.**

lab 5. Binary Search Tree

- ## Input
  - Obtain a list of operations from the given input file, and execute the given operations in order.
  - A detailed specification of the operations is provided below.
  - Each line represents a single operation.
  - Each operation and the necessary parameters are separated by a space.
  - Input values (represented as x below) are any integer.
- ## Operations
  - **i x** : insert a new value “x” into the binary search tree without duplication. If x already exists in the tree, print message.
  - **d x** : delete a value “x” in the binary search tree. If x does not exist in the tree, print message.
  - **f x** : Find the given value to check whether the key exists in the tree.
  - **pi** : print the tree by inorder traversal
  - **pr** : print the tree by preorder traversal
  - **po** : print the tree by postorder traversal

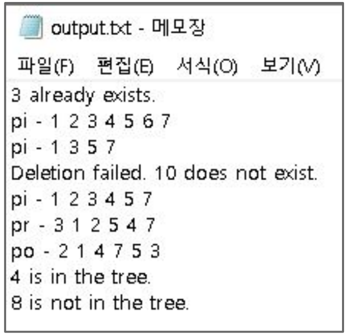
lab 5. Binary Search Tree

- **Input**



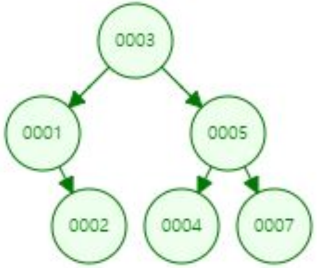
```
i 4
i 2
i 6
i 1
i 3
i 5
i 7
i 3
pi
d 4
d 2
d 6
pi
i 4
i 2
d 10
pi
pr
po
f 4
f 8
```

- **Output**



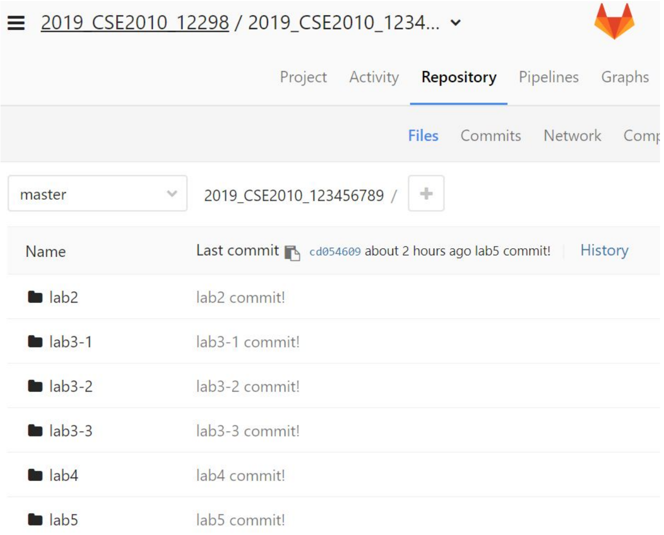
```
3 already exists.
pi - 1 2 3 4 5 6 7
pi - 1 3 5 7
Deletion failed. 10 does not exist.
pi - 1 2 3 4 5 7
pr - 3 1 2 5 4 7
po - 2 1 4 7 5 3
4 is in the tree.
8 is not in the tree.
```

- After executing the last command, the tree should look like this. (use FindMax in the left subtree during DeleteNode operation).

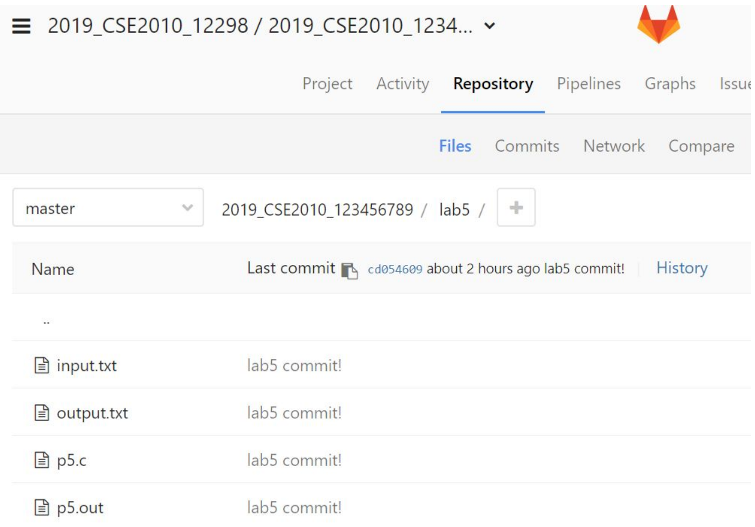


lab 5. Binary Search Tree

- Submission
  - Project directory name : lab5
  - Source file name : p5.c
  - Executable file name : p5.out
  - You should upload in the hconnect (Gitlab) server.



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!



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, 10 April, 23 : 59 pm