HW5: Binary Search Tree

제출은 gitLab을 통해서 하며 **http://hconnect.hanyang.ac.kr/2017\_CSE2010\_수업번호/2017\_CSE2010\_수업번호\_학번.git 에 HW5라는 폴더를 만들어 진행.**

\* 프로그램 제출간 유의사항

- 소스코드에는 주석이 있어야 함.

- 주어진 구조체와 input.txt를 사용해야 함

**- 숙제의 소스코드 평가는 linux ubuntu 16.04.2 LTS 버전 gcc 5.4.0에서 함.**

\* 보고서 제출간 유의사항

- 작성한 소스 코드가 첨부되어야 하며, 실행결과가 첨부되어야 함.

- 분량은 제한이 없으나 1~2apge로 간략하게 설명.

- 보고서는 hw5\_학번.확장자(doc, docx, pdf)로 제출.

**제출시간: '17.4.19(23:59) 까지**

**\*지연제출**

- 24시간 이내는 해당 과제 50% 감점, 48시간 이내는 75% 감점.

- 지연제출자는 E-mail(casualab@hanyang.ac.kr)과 gitlab에 모두 제출.

- E-mail제목: "hw5\_학번\_자신의 수업 요일(수, 목)\_이름 " 형식으로 제출.

We will implement binary search tree ADT with the three main functions using **iteration**: insert, delete, and search.

* Input

Obtain a list of operations from the given input file, and execute the given operations i 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. You may assume that the input values (represented as x below) are any integer.

* **i x**: insert a new key “x” into the binary search tree without duplication.
* **d x**: delete a key “x” in the binary search tree. If x does not exist in the tree, print an error message "key is not in the tree".
* **s x**: find the given key to check whether the key exists in the tree
* **p**: print the entire tree. Use the given “display function”.

An input file is shown below.

|  |
| --- |
|  |

* Binary Search Tree ADT

(1) Data Specification for the objects

typedef struct TreeNode {

int key;

struct TreeNode \*left, \*right;

} TreeNode;

(2) Function specification

- void insert\_node(TreeNode \*\*root, int key)

-Insert a new node with the key value into the tree.

- void delete\_node(TreeNode \*node, int key)

-delete a node with the given key value from the tree.

-If the key does not exist in the tree, print an error message “key is not in the tree: %d”.

-If the node to be deleted has both children, it is replaced with the smallest node in the

right subtree.

- TreeNode \*search(TreeNode \*node, int key)

- Find the key in the binary search tree.

- Print “key is in the tree: %d” if the key exists.

Otherwise, print “key is not in the tree: %d”.

3. Program description

* name : hw5\_학번.c
* input : a list of operations in a file (an input file name is given as a command line argument. See the example in “1. input” on the first page)
* output : the corresponding result in the standard output