HW6: AVL tree

제출은 gitLab을 통해서 하며 **http://hconnect.hanyang.ac.kr/2017\_CSE2010\_수업번호/2017\_CSE2010\_수업번호\_학번.git 에 HW6라는 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 hw6\_학번.확장자(doc, docx, pdf)로 제출.

**제출시간: '17.5.17(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: "hw6\_학번\_자신의 수업 요일(수, 목)\_이름 " 형식으로 제출.

We will implement AVL tree ADT, in each node of the AVL tree, the difference between the height of the left subtree and the height of the right subtree is at most 1. We will implement *avl\_add* function, single rotation(rotate\_right, rotate\_left), double rotation(rotate\_right\_left, rotate\_left\_right). Additionally, students who implement the 'avl\_delete' function will receive additional points.

1. Input

Obtain a list of numbers from the given input file, and execute an insertion operation for each number in order. At each iteration of insertion, print the AVL tree inorder traversal. An example input file is shown below.

Input.txt

|  |
| --- |
| Result: |

2. AVL tree ADT

(1) Data Specification for the objects

struct avl\_node {

struct avl\_node \*left\_child, \*right\_child; /\* Subtrees. \*/

int data; /\* Pointer to data. \*/

};

struct avl\_node \*root;

(2) Function specification

* struct avl\_node\* rotate\_right(struct avl\_node \*parent)

- This function is used to rebalance the left left unbalanced case.

* struct avl\_node\* rotate\_left(struct avl\_node \*parent)

- This function is used to rebalance the right right unbalanced case.

* struct avl\_node\* rotate\_right\_left(struct avl\_node \*parent)

- This function is used to rebalance the right left unbalanced case.

* struct avl\_node\* rotate\_left\_right(struct avl\_node \*parent)

- This function is used to rebalance the left right unbalanced case.

* int get\_height(struct avl\_node \*node)
* int get\_height\_diff(struct avl\_node \*node)
* struct avl\_node\* rebalance(struct avl\_node \*\*node)
* struct avl\_node \* avl\_add(struct avl\_node \*\*root, int new\_key)
* struct avl\_node \* avl\_delete(struct avl\_node \*\*root, int new\_key)

- Students who implement this function will receive additional points.

* void display(struct avl\_node \*node)

3. Program description

* name : hw6\_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