HW7: B-tree

제출은 gitLab을 통해서 하며 **http://hconnect.hanyang.ac.kr/2017\_CSE2010\_수업번호/2017\_CSE2010\_수업번호\_학번.git 에 HW7라는 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 hw7\_학번.확장자(doc, docx, pdf)로 제출.

**제출시간: '17.5.24(23:59) 까지**

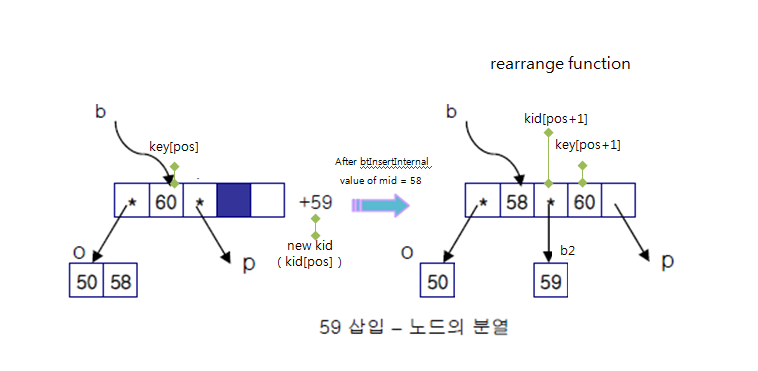
**\*지연제출**

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

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

- E-mail제목: "hw7\_학번\_자신의 수업 요일(수, 목)\_이름 " 형식으로 제출.

We will implement b-tree ADT(MAX\_KEYS: 2 or 4)'s btInsert function. The btInsert has a btInsertInternal function with subroutine(rearrange function). The rearrange subroutine changes the b-tree like below. Please refer to given.c file.



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, BtPrintKeys function is already implemented and you have to use it. An example input file is shown below.

Input1.txt (MAX\_KEYS 2)

|  |
| --- |
| i 50  i 58  i 60  i 59  Result: |

Input2.txt (MAX\_KEYS 4)

|  |
| --- |
| i 10  i 20  i 5  i 6  i 12  i 30  i 7  i 17  Result: |

2. AVL tree ADT

(1) Data Specification for the objects

#define MAX\_KEYS (2) // When using input2: #define MAX\_KEYS (4)

struct btNodeInternal {

int numKeys; /\* how many keys does this node contain? \*/

int keys[MAX\_KEYS];

struct btNode \*kids[MAX\_KEYS+1];

};

struct btNode {

int isLeaf; /\* is this a leaf node? \*/

btNodeInternal keysAndKids;

};

typedef struct btNode \*bTree;

(2) Function specification

* bTree btCreate(void)

- This function is used to create a new empty tree.

* void btDestroy(bTree t)

- This function is used to free a tree.

* int btSearch(bTree t, int key)

- Return nonzero if key is present in tree. It has a searchKey subroutine.

* void btInsert(bTree t, int key)

- This function is used to insert a new element into a tree. It has a btInsertInternal subroutine.

* void btPrintKeys(bTree t)

- Print all keys of the tree in order.

* void rearrange(bTree b, bTree b2,int pos, int mid)

- This function changes the shape of the tree after the btInsertInternal subroutine.

Please refer to given.c file and the above picture.

3. Program description

* name : hw7\_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