**코드 명세서**

**B-TREE**

컴퓨터소프트웨어학부

2019040591

박진수

**개발 및 실행 환경**

OS: Windows\_NT x64 10.0.19045 언어 버전: C17

컴파일 환경: gcc (x86\_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0 IDE 종류: VS code 1.87.0

**코드 설명**

typedef struct \_TreeNode {

  int keys[M];

  int value[M];

  int numOfKeys;

  struct \_TreeNode \*children[M + 1];

  int numOfChildren;

  bool isLeaf;

} TreeNode;

B-tree의 각 노드를 구성하는 구조체입니다. 해당 노드의 key, value, key의 개수, 자식 포인터들의 배열, 자식 포인터의 개수, 그리고 해당 노드가 리프노드인지 여부를 저장합니다.

TreeNode \*createNewNode(int *key*, int *value*);

key와 value를 입력으로 받아 해당 값을 가지는 노드를 생성하는 함수입니다. 구조체의 크기만큼 메모리를 할당하고, struct가 필요로 하는 값들을 초기화합니다. 해당 노드의 포인터를 return 합니다.

TreeNode \*splitNode(int *splitIndex*, TreeNode \**node*, TreeNode \**parent*);

노드가 가질 수 있는 key값의 최댓값을 넘어섰을 때 실행되는 함수로, 가운데(median) 노드를 기준으로 좌우로 분할한 후, 가운데 값을 부모 노드로 승진시키는 함수입니다.

TreeNode \*insertData(int *key*, int *val*, int *parentIndex*, TreeNode \**currentNode*, TreeNode \**parent*);

집어넣으려고 하는 key값과 value 값을 해당 subtree에 집어넣는 함수입니다. 적절한 leaf노드의 위치에 삽입한 후, 재귀함수를 빠져나오면서 가득 찬 경우 split 연산을 실행합니다.

void insert(int *key*, int *val*);

전역변수인 root에 해당 key와 value를 삽입합니다. root가 NULL인 경우, 해당 값을 data로 가지는 새로운 노드를 생성한 후, root로 삼고, 기존 tree가 존재할 경우 insertData함수를 실행합니다.

bool searchNode(FILE \**pFile*, TreeNode \**node*, int *key*);

btree를 순회하며 해당 key를 가진 노드의 data를 찾습니다. 만약 발견할 경우 전달받은 FILE 포인터를 통해 해당 key값과 value값을 파일에 write합니다.

char \*getNewFileName(char \**fileName*);

입력으로 받는 파일명을 입력으로 받으면, 새롭게 생성된 파일명을 return하는 함수입니다. 예를 들어 “input.csv” 파일명을 인수로 받으면 “input\_created.csv”를 반환합니다. 이처럼 중간에 “\_created”를 추가해줍니다.

void compareTwoFiles(FILE \**file1*, FILE \**file2*);

두 파일이 서로 같은 내용을 가지고 있는지 비교하여, 정확히 일치하는지 여부를 콘솔창에 출력해주는 함수입니다.

int main();

메인 함수의 경우, 화면 입출력과 파일 입출력을 담당하도록 구현하였습니다. 1,3의 숫자를 입력 받아 해당하는 기능이 동작하도록 하였으며, 입력 받을 파일명을 입력하면 해당 파일을 한 줄씩 읽고 비교하거나, 새로운 파일을 생성하여 해당 파일에 key와 value pair을 write하도록 했습니다.

감사합니다.