

숙제: B-tree 개발

- 숙제 내용:

B-tree 프로그램을 구현하고 실험한다. 주어진 데이터를 B-tree 에 모두 저장하고, 여기의 레코드를 탐색, 삭제 해 보는 작업을 실험해 본다. B-tree 노드의 capacity order d 는 다음처럼 상수로 선언되도록 하고 이 d 를 이용하여 모든 변수의 타입을 선언하도록 하고 프로그램에서도 이 d 를 이용토록 한다. (capacity order d = 2 로 해서 실험 해 볼 것).

- 삽입할 데이터는 두 개의 텍스트 파일 Com_names1.txt , Com_names2.txt 에 들어 있는 업소명들이다. 하나의 업소명은 하나의 레코드에 저장된다. 레코드는 업소명의 문자열과 업소명의 길이의 두 개의 필드를 가지며 다른 필드는 없다. 따라서 하나의 레코드의 정의는 다음과 같다:

```
typedef struct element{    // 레코드 정의. 회사명과 그 길이로 구성됨
    char name[100]; // 회사명
    int nleng; // 회사명 길이
}ele;
```

- 주의: 이 과제에서 key 값은 이름 즉 문자열이다. 문자열 간의 비교는 사전식 순서에 의거하여 비교하도록 한다 (strcmp 함수 사용).

- 작업 수행 순서:

(1) 저장 단계:

주어진 두 개의 회사명 파일 안의 모든 회사명을 B-tree 에 저장한다 (주: 중복으로 저장되지 않도록 한다). 방법은 하나의 회사명을 저장하는 함수 insert_arec 을 작성하고 이를 반복적으로 호출하여 모든 회사명을 저장하도록 한다. 이 함수는 입력으로 받은 문자열에 대하여 레코드를 만들고 이 레코드를 B-tree에 삽입한다.

int insert_arec(...){ ... } 에서 **/* Fill your code */** 부분을 채운다.

```
/*
Fill
your
code
*/
```

와 같이 5줄로 표시된 부분은 여러 줄을 의미한다.

(2) 테스트 단계:

프로그램을 실행하면 먼저 주어진 데이터 파일 2개 안의 모든 줄의 상호명을 b-tree 에 삽입하는 단계를 수행한다. 그 다음 단계에서는 특정 레코드의 삽입(insertion), 삭제(deletion), 탐색(retrieval), 순차출력(sequential scanning) 에 대하여 실험한다.

void redistribution(...) { ... }에서 **/* Fill your code */** 부분을 채운다.

int B_tree_deletion(...) { ... }에서 **/*Fill your code */** 부분을 채운다.

순차출력은 저장된 모든 레코드들은 키의 오름차순으로 출력하는 것이다 (한 줄에 한 레코드 출력).

실험 예;

프롬프트 "명령을 넣으시오" 에 대하여 해당 작업을 수행하는 루프를 반복한다.

명령을 넣으시오

insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> r 동음농협
→ 해당 이름을 찾은 다음 발견된 위치(레벨 및 레코드 내의 위치)를 출력한다.

명령을 넣으시오

insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> d 동음농협
→ 해당 이름을 제거한 다음 성공 여부를 출력한다 (그리고 이 과정에서 merging이 발생할 때 마다 "Merging has been done."을 출력한다).

명령을 넣으시오

insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> p
→ B-tree 안의 모든 레코드를 이름 순으로 출력한다.

명령을 넣으시오

insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> i 연세생협
→ 주어진 상호명을 가지는 레코드를 만들어 b-tree 에 삽입한다.

명령을 넣으시오

insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> e
→ 프로그램을 종료한다

(3) 제출:

B_tree_proj_2022_guid.c 파일을

HW#1_학번_이름.zip 파일명으로 압축하여 제출한다.

```
C:\D:\PROGRAMS\TEACHING\Fundamentals_Algorithm_2022\B_tree_proj_2022.exe
삽입 성공한 레코드 수 = 40215

명령을 넣으시오
insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> i 간성수
삽입 성공.

명령을 넣으시오
insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> r 간성수
탐색성공. Name: 간성수, 길이: 6

명령을 넣으시오
insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> d 동읍소방
Merging has been done.
Merging has been done.
Merging has been done.
Merging has been done.
삭제 성공.

명령을 넣으시오
insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> r 동읍소방
탐색요청 받은 키를 가진 레코드가 존재하지 않음.

명령을 넣으시오
insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> i 천덕군
삽입 성공.

명령을 넣으시오
insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >> r 천덕군
탐색성공. Name: 천덕군, 길이: 6

명령을 넣으시오
insert: i 이름 / delete : d 이름 / retrieve : r 이름 / 전체출력: p / 종료: e >>
```