

숙제: B+tree 개발

- 숙제 내용: B+ 트리를 이용하여 데이터를 저장하고 이를 검색하는 실험을 수행한다.

Index structure 의 capacity order = 2,

Data node 의 capacity order = 2 로하여 시스템을 개발한다.

- 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 record{      // 레코드 정의. 회사명과 그 길이로 구성됨
    char name[100]; // 회사명
    int leng; // 회사명 길이
}type_rec;
```

- 주의:

- 이 과제에서 key 값은 이름 즉 문자열이다. 문자열 간의 비교는 strcmp 함수를 사용하여 사전식 순서에 의거하여 비교한다.
- 데이터 노드들은 sequence set이라 불리는 연결리스트로 연결한다. 첫 데이터 노드를 가리키는 포인터를 전역 변수 HEAD에 저장한다.
- 인덱스 구조의 루트 노드에 대한 포인터는 전역 변수 ROOT가 가지도록 한다.
- 부모 정보를 저장하는 스택은 전역 변수 stack과 top으로 구현한다.

- 작업 수행 순서:

(1) 저장 단계:

주어진 두 개의 회사명 파일 안의 모든 회사명을 B+tree 에 저장한다.

(2) 테스트 단계:

프롬프트 "명령> " 에 대하여 명령을 넣어 해당 작업을 수행하는 루프를 반복한다.

수행할 명령들: se, sp, rs, ex

명령> se *탐색키*

해당 키를 가진 레코드를 탐색하여 모든 필드를 출력한다.

명령 > sp

B+ tree 안의 모든 레코드들을 키의 오름차순으로 sequential_print.txt 파일에 출력한다(한 줄에 한 레코드).

[방법]: HEAD 가 가리키는 데이터노드에서 출발하여 연결리스트 링크(link) 를 따라 가면서 만나는 모든 데이터노드의 레코드를 출력한다.

명령 > rs 시작키 최종키

시작키에서 출발하여 최종키까지 속하는 모든 키를 가진 레코드들을 파일에 출력한다(한 줄에 한 레코드).

출력 파일명: range_search_output.txt.

[방법]: 시작키가 들어 있을 데이터노드에 대한 포인터(ptr1)를 구한다. 그리고 최종키가 들어 있을 데이터노드에 대한 포인터(ptr2)를 구한다. ptr1 노드 내부를 스캔하여 시작키 및 그 이후의 모든 키를 가진 레코드를 출력한다. ptr1과 ptr2 사이의 모든 노드들을 하나씩 방문하여 내부의 모든 레코드들을 출력한다. ptr2 노드에서 내부 레코드들 키가 중 최종 키보다 작거나 같은 모든 레코드를 출력한다.

(주의: 시작키보다 최종키가 크면 안 된다. 그리고 ptr1과 ptr2가 동일 노드인 경우도 처리되어야 한다.)

명령 > ex

프로그램을 종료한다

- 샘플 코드 제공: 원래 개발은 처음부터 자신이 작성하는 것이 가장 좋다. 그러나 개발에 도움을 주기 위해 샘플코드를 제공한다. 제공된 샘플코드 내 **/* Fill your code */** 부분을 채워서 개발을 진행하는 것도 가능하다.

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

 와 같이 5줄로 표시된 부분은 여러 줄을 의미한다.

- 제출:

- B_plus_tree_2022_guide.c 파일
- 실험 상황을 보여주는 실행창 캡처 이미지를 넣은 보고서 (한글, 워드로 작성)를

HW#2_학번_이름.zip 파일명으로 압축하여 제출한다.