**데이터베이스[CSE3207-003] Project 2: B+-tree**

컴퓨터공학과 3학년

12171645 심규범

**What you have implemented and what you have not**

insert, search(point), print, main 함수들 그리고 insert 구현 중 필요해진 함수 read\_block을 구현해보려 했으나, insert 중 non-leaf node를 입력하는 것에 어려움을 느껴 모든 함수들에 대해 leaf node만 다룰 수 있는 한에서 구현하였습니다. Range search에 대해서는 어떻게 설계를 해야할 지 몰랐으며, 나머지들은 설계해봤지만 원래의 의도와 목적에 따라 구동하는 함수는 없었습니다.

**Brief explanation of your implementation**

|  |  |
| --- | --- |
|  | **Class Entry**  Key, Value, 그리고 사용되었는지 아닌지를 표시하는 fill로 이루어져 있다. |
|  | **Class Node**  Next\_bid: non leaf node 와 leaf node를 가리키는 block id  B: 노드의 max entry + 1  Max entry: 노드의 최대 entry 개수  Count: non leaf node와 leaf node를 구별해주기 위해서 사용하려 하였으나 사용하지 않음  Insert\_entry: entry의 특정 위치에 node를 삽입하기 위한 기능 |
|  | **Class Btree**  Block\_size: 블록의 사이즈  Root\_bid: root block id  Last\_bid: last block id  Depth: 전체 트리의 깊이  B: 노드의 최대 entry 개수 + 1  Max\_entry: 노드의 최대 entry 개수  Fstream file: file I/O  FILE\* pFile: fopen, fwrite, fread, fclose들을 사용할려 했으나 코드가 꼬여서 fstream file로 변경함  Root: root node  File\_name: I/O할 file의 이름  Read\_block(): bid가 속해져 있는 block을 읽어옴  Insert(): node들에 data 삽입(leaf node에 한해서 구현함)  search(), point: leaf node에서entry의 value를 return  search() range: 해당 범위 내의 node의 entry의 value들을 return 하려 했으나 구현하지 못함  Print: 모든 node들을 출력하나 구현하지 못함 |
|  | **Main 함수**  Case에 따라 작동 |

**How to compile and run**

Visual Studio 2019에서 컴파일(ctrl+F5 or F5) 및 실행하였습니다.

**Talk about your experience of doing this project**

B+ tree의 구조를 생각하기 이전에 File 입출력에 대해서 아는 것이 거의 없어서 많은 어려움을 겪었습니다. File 입출력을 다룰 줄을 모르기에 B+tree알고리즘 자체에 대해서 고민해보는 시간보다 File 입출력에 대해서 구글링하고 공부하는데 시간을 더 많이 쏟았습니다. 처음에는 막무가내로 project2\_tutorial에서 소개된 FILE\* 메소드들(fopen, fseek, fwrite, fclose 등)을 사용해봤으나 코드를 작성하면 할수록 어려움이 생겨서 후에 1학년 때 배웠던 fstream을 사용하여 코드를 짰습니다. Fstream을 사용하니 코드상의 오류도 줄고 사용하기가 편했습니다. 사용했던 FILE 메소드들은 주석처리 하여서 코드에 남겨놓았습니다.

그리고 sample case들을 보면서 이것들을 어떻게 사용하는건지, 그리고 file의 생성은 어떻게 하는건지 감도 안 잡혔습니다. Project2\_description을 보면서 command-line parameter을 어떻게 사용하는 건지 몰라서 어려움을 겪었습니다. File을 cin과 cout을 사용하지 않고 어떻게 입출력하는지 몰라서 구글과 책을 많이 찾아봤습니다. 그와 동시에 이전에 해본 적 없는 코딩의 세계에 온 듯하여서 신기하기도 하였으나 실사용은 헷갈렸습니다. 이렇게 B+tree의 알고리즘 자체보다는 코드 구현 단계에서 많은 시간을 소요하여 정작 B+tree 구현에 원했던 만큼의 시간을 투자하지 못했습니다. 그리하여 leaf-node 단계들에 한해서 함수들을 구현해봤습니다. 더 많은 시간이 있었으면 non-leaf node와 next-level node를 고려해서 B+tree를 더 많이 구현할 수 있지 않았을까 하는 아쉬움이 남습니다.

Project 1에서 SQL Query에 관련해서 과제를 진행하였기에 Project 2는 상대적으로 저번학기에 수강한 자료구조 과제를 하는 느낌이였습니다.

**contact information**

12171645 심규범, 01027095758, koreanshim@naver.com