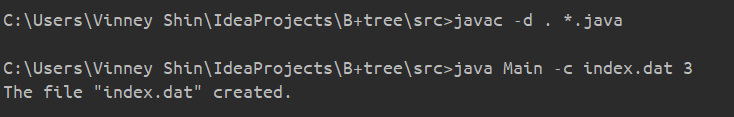
2018 B+tree implementation

Database systems ITE 2038-12521 김상욱 교수님

2017029870 신호중

1. Summary of the algorithm
   1. Insertion
      1. Insertion을 수행할 node를 찾고, 그것이 루트인지 확인합니다. 만약 루트이면서 키 개수가 0개라면 특수 상황이므로 0번째 인덱스에 key와 value를 적절히 추가해줍니다. 만약 아니라면 그 키 값보다 작은 노드를 찾아가 알맞은 인덱스에 키와 벨류를 넣어줍니다.
      2. 삽입 후 항상 오버플로우가 일어났는지 체크합니다. 오버플로우가 일어나지 않았다면 그대로 두고, 오버플로우가 일어났다면 중간 인덱스 노드를 기준으로 분열시키고 그 노드의 값을 부모로 놓습니다.
      3. 이것을 모든 부모노드들의 분열이 끝날 때까지 계속합니다
   2. Deletion
      1. 시간이 부족해 짜지 못했습니다. 죄송합니다 하지만 deletion을 위한 left, right sibling making까지는 완벽히 구현해 놓았습니다.
2. Compile environment
   1. JDK-9.0.4를 사용했습니다.
   2. cmd에서 직접 컴파일해서 사용하는 경우 java 폴더 안 java파일들을 컴파일 하신 후에 아래 사진과 같은 형식으로 사용하시면 됩니다.



1. Description of my code
   1. Node.java (LeafNode와 InternerNode의 abstract parent class 이다)

Leaf와 Nonleaf 노드의 추상클래스로 상속시켜 사용하였습니다. 오버플로우가 일어날 시 분열시킬 수 있는 메소드인 dealwithOverFlow와 split을 가지고 있고 split와 search는 abstract method로 Leaf 와 InternerNode에서 다르게 작동합니다. 이외 적절히 Rightsibling 과 Leftsibling, key parent 등을 얻을 수 있는 getter, setter method들도 존재하며, 혹시모를 사용에 대비해 compare을 override 하는 sort함수도 구현해 놓았습니다. Keys를 가지고 있는 arraylist를 가지고 있습니다.

* 1. BTree.java

기본적인 기능인 insert, delete, single key search, range key search 에 대한 기능을 구현해 놓았으며 key를 저장할 알맞은 node를 찾는 findRightLeafNode method가 구현되어 있습니다. 이 method는 single search와 range search에 대해 동작할 수 있도록 구현해 놓았습니다.

Insert는 기본적으로 파일입출력의 bufferedReader와 StringTokenizer을 적절히 섞어 ,를 기준으로 저장된 key와 value를 parsing하여 사용하도록 구현해 놓았습니다. findRightLeafNode를 통해 root를 베이스로 적절한 leaf노드를 찾은 뒤 LeafNode class에 구현된 insert 메소드를 통해 key와 value를 집어 넣습니다.

아쉽게도 delete는 모두 짜지 못했습니다.

* 1. FileManager.java

FileManager클래스는 createFile, saveFile, loadFile로 이루어져 있습니다. PrintWriter을 사용했으며, leafNode를 차례로 key value pair로 index.dat에 저장했습니다. 이름이 같다면 기본적으로 append가 되도록 구현해두었습니다.

* 1. LeafNode.java

InternerNode와 다르게 작동하는 split을 가지고 있으며 rightLeafNode와 rightSibling 을 적절히 사용하여 Leaf노드 수준에서 traversal이 가능하도록 만들었습니다. 이외 search, insert를 가지고 있으며, insert는 root노드인지, 오버플로우가 일어났는지에 따라 findRootNode메소드를 사용해 분열합니다. Value를 가지고 있는 arraylist가 존재합니다.

* 1. InternalNode.java

마찬가지로 LeafNode와 다르게 작동하는 split을 가지고 있으며 (중간 값을 삭제하는) rightsibling과 leftsibling을 deletion을 위해 잘 연결되도록 구현해두었습니다. Child에 대한 getter setter 가 있습니다. Child에 대한 arraylist를 가지고 있습니다. pushUp과 insertAt 함수를 통해 왼쪽, 오른쪽 child와 가운데 키값을 적절히 넣도록 만들었습니다.