2013012148 이재일

B+ tree 구현 보고서

우선 9.21일 저녁까지 비트리를 구성하는데 성공했지만 시간이 3시간밖에 남지 않아서 eclipse로만 구현되는 b+ tree를 만들었습니다.(사용자의 입출력 부분을 구현이 미완성입니다..)

제약조건 : degree는 4이상으로 해야 합니다. (3일때는 outofbound error발생.. 중간에 half-2 때문에 4 미만으로 할 경우 -1이 되기 때문입니다ㅜㅜ. 시간이 없어서 고칠 수가..)

주석을 자세히 달아서 코드만 보아도 이해가 되도록 작성했습니다. 보고서는 주로 흐름 위주로 작성하겠습니다.

1. abstract Class Node

이 클래스는 LeafNode와 IndexNode가 상속을 받는 추상 클래스이다. 나중에 Bplustree 클래스의 root의 맴버변수로 선언되어서 Leaf와 Index 모두를 받을 수 있게 한다.

root노드가 초기 split 전에는 leaf node, split 후에는 index node가 된다. 따라서 node를 추상 클래스로 선언한 후 leaf node와 index node가 node 클래스를 상속받도록 하여서 node로 선언한 root가 leaf와 index 모두를 가리킬 수 있도록 한다.

2. LeafNode class

Node를 상속받는다. pair로는<int,int>형으로 key와 value를 가진다.

맴버변수 l(left)과 r(right)로 형제노드와 왼쪽 오른쪽 노드간의 이동이 가능하다.

맴버변수 parent로 부모노드(무조건 indexnode가 된다)로 이동이 가능하다.

3. IndexNode class

Node를 상속받는다. pair로는 <in,Node> 형으로 key와 node를 가진다.

맴버변수 r을 통해 rightmost child로 이동할 수 있다.

맴버변수 parent로 부모노드로 이동이 가능하다.

4. Bplustree class

b+ tree의 연산을 모아놓은 class이다.

5. bptree class

main 메소드를 가지는 class. 원래는 여기서 입출력이 수행되어야 하는데 시간이 없어서 구현을 못했다. 콘솔로 간단히 test하는 class가 되어버렸다.

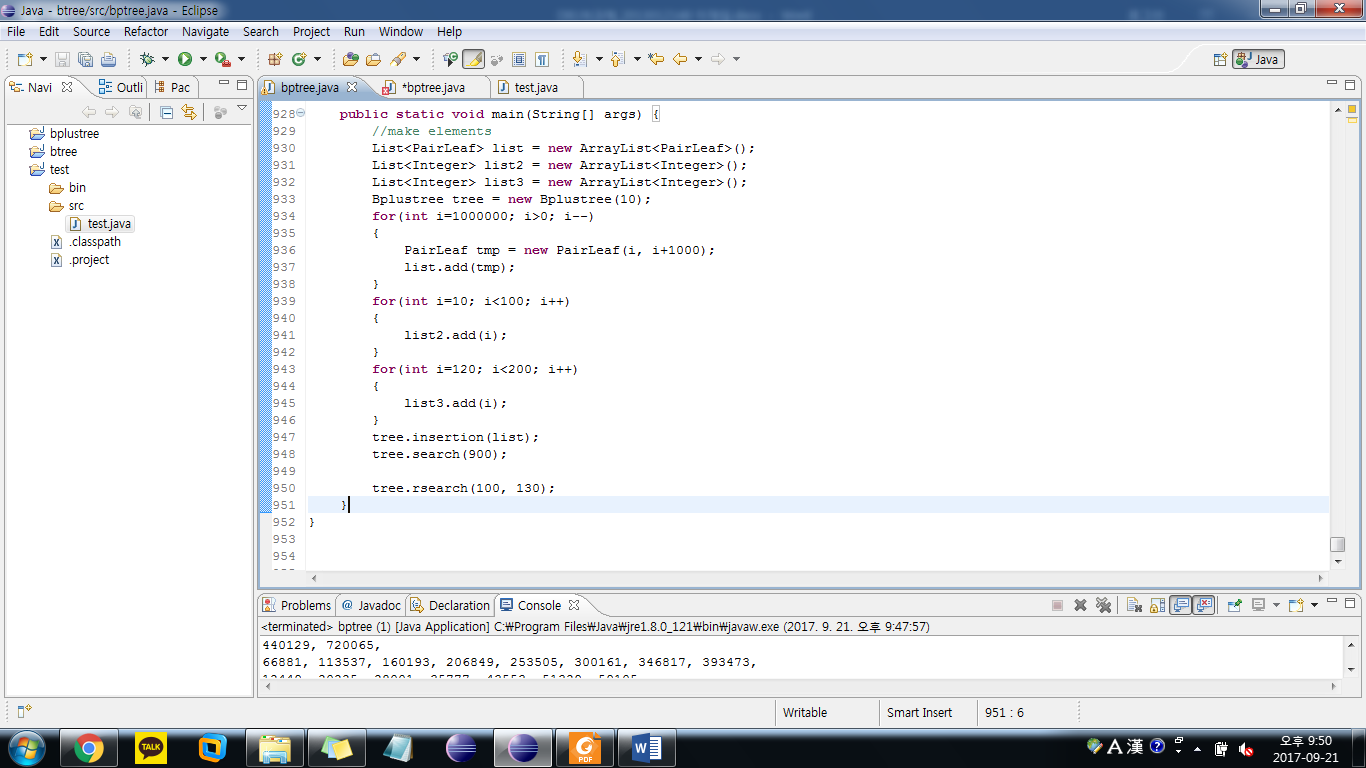
6. insertion 연산.

우선 root에서 searchLeaf 메소드를 통해서 해당 key가 삽입되야할 leaf 노드를 받는다. 해당 leaf 노드에서 <key, value>를 insert하는데 만약에 full일 경우 split을 진행한다. 그리고 split을 통해서 부모 노드로 올라가는 key는 parent 노드에 insert을 한다. split을 통해서 root가 변하는 경우를 대비해서 Bplustree class가 insert메소드에 인자로 전달되며 setroot를 통해서 root를 변경시켜준다. index노드에서 해당 key가 insert될 때 해당 insert노드가 full이 아니면 종료시킨다. 만약 full일 경우 insert node split을 하게 된다. insert split은 해당 key가 부모노드로 올라갈 때 현재 노드에서 제거해준다. 위와 같이 부모노드가 계속 full일 경우 연속적으로 split이 된다.

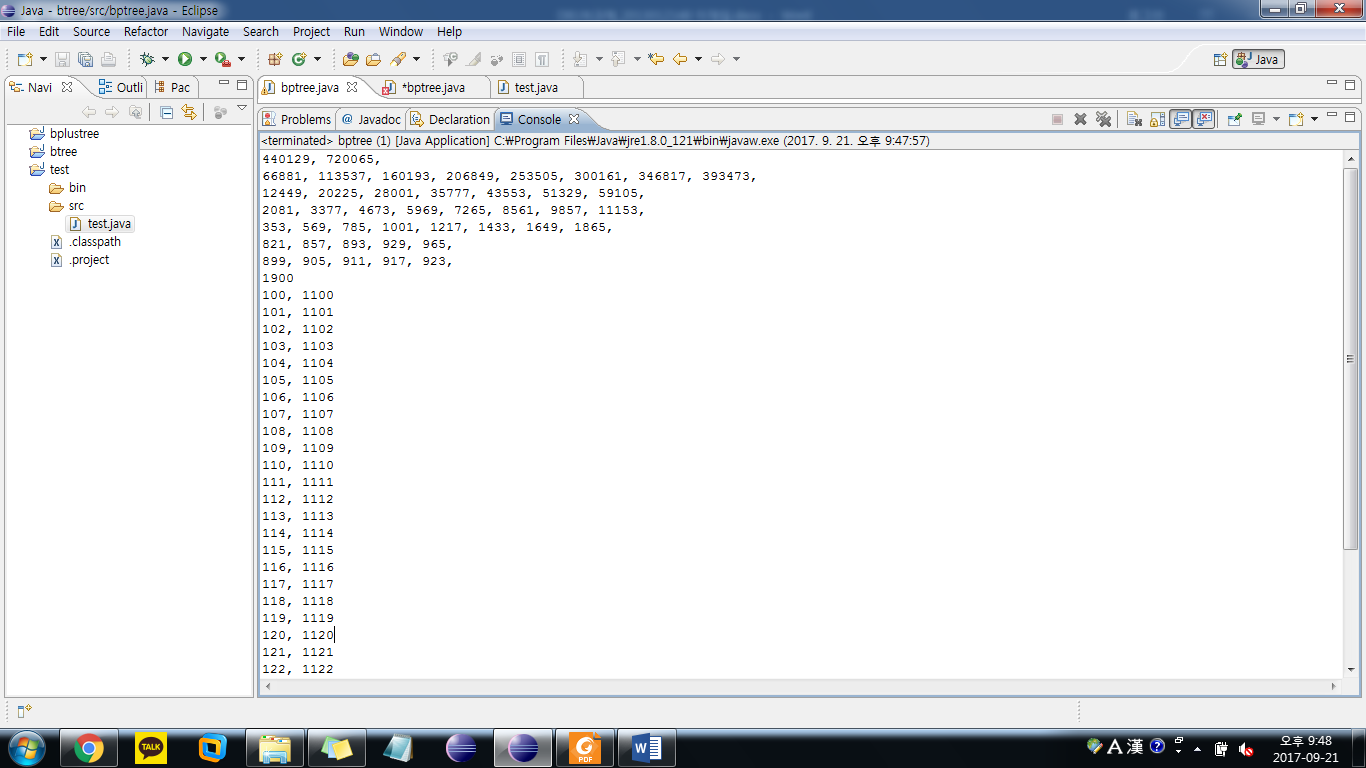
7.deletion

우선 root에서 searchLeaf 메소드를 통해서 해당 key가 존재할 수 있는 leaf노드를 받는다. 해당 leaf node에서 key가 존재하지 않는 경우 key가 없음을 화면에 표시하고 종료한다. key가 존재할 경우 삭제해준다. 삭제 후 해당 leaf node의 원소가 half보다 적다면 왼쪽 형제 노드에서 원소 1개를 borrow한다. 이경우 왼쪽 형제 노드가 빌려준 노드의 key로 기존 두 노드 사이의 key가 변경된다. 만약 왼쪽 노드도 half일 경우 오른쪽 노드가 half가 아닐 경우 오른쪽 형제 노드에서 원소 1개를 borrow한다. 만약 두 노드 모두 half일 경우 왼쪽 노드와 merge한다. 만약 왼쪽 노드가 존재하지 않는다면 오른쪽 노드와 merge한다. 만약 왼쪽 노드와 오른쪽 노드 모두 없다면 해당 노드는 root노드이다. merge를 하는 경우 해당 노드 사이의 key가 부모노드에서 삭제가 되는데 이 경우 부모 노드의 원소가 half일 경우 다시 위의 (왼쪽 borrow 가능? -> 오른쪽 borrow가능? -> 왼쪽 merge? -> 오른쪽 merge?)를 반복하게 된다.

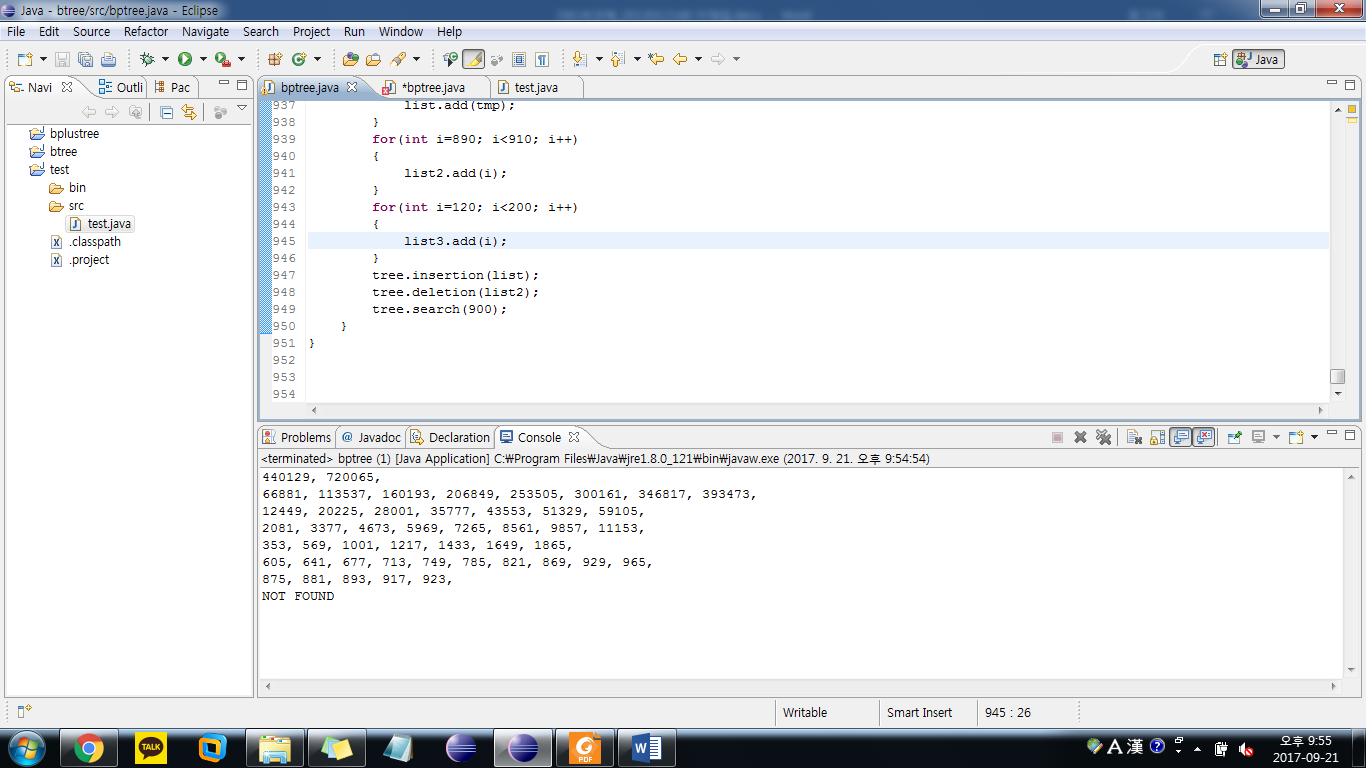
7. 수행결과



입력값으로 100만개의 input(for문으로 key값 i, value값 i+1000)을 넣은 후 단순 search로 900을 찾고 100부터 130까지 rsearch를 수행하는 main입니다.



해당 main을 수행한 결과입니다.



삭제연산을 진행한 결과입니다.