

2020-Summer-CSE-5331-002-Software Testing
Prof. Sharma Chakravarthy
Project 1 - Team 11
Jay Denton, ID - 1001559386

Overall Status

All the three functions: insert(), _insert() and NaiveDelete() has been implemented and can be compiled with no errors shown (not sure about details, but to me, seems it works).

The Makefiles are remained as the original.

File Description

There are no new files. The javaDoc already exists. There are comments written by myself line to line. (Quite a few lines of comments).

Division of Labor

By myself. And questions answered by TA and the Prof. (Thank y'all :)

Logical Errors and How I Handled Them

1. (This is not quite an error. But I modified it, anyway.) The “flag” variable in the BTDriver class within BTTest.java file is set 0 at first. So when I try to insert the first key into an empty tree, I cannot have an idea whether my insert() function runs correctly or not.

Solution: I initialized it to be flag = 1. So I can know exactly how the first key is inserted into the tree.

2. After the first split of the first leaf page in the B+ tree, the index file is created, but somehow, it doesn't show the tree level correctly.

Solution: I found out that the headerPage type is still (integer)12, which is LEAF. Thus, I added the step to change the headerPage type to be INDEX-11, after the split.

3. In the _insert() function, when the leaf page or the index page is full, I need to split them correctly, however, after two loops happened, the printout is showing incorrect leaf information (e.g. 0 (key, [pageno slotno]) : (10, [3 0]), or something like that).

Solution: It turns out that I shouldn't use delRid variable directly, but instead, I can use “(LeafData)upEntry.data).getData()” which returns the correct RID.

4. In the NaiveDelete() function, I was trying to follow the procedure which described in the demo file, but seems like its a combination of the delete() function, findRunStart() function and the NaiveDelete() function.

Solution: I used the while loop, with fewer lines just to loop though the founded leaf page, where the key exists. Using “delEntry = leafPage.getNext(delRid);”. The loop won't stop until the spot of the key is found.