Project 1 Report

Overall Status

Current status: *In QA*

Implemented Correctly

Insert(), _insert(), and NaiveDelete() are believed to be working properly

Unfinished Work

None.

File Descriptions

No new files were created. Only changes are in the designated area in BtreeFile.java.

Division of Labor

Initially we divided the labor by getting together and completing the required functions in the order which they are needed. This is Insert(), _insert(), then NaiveDelete(). This work was done in a pair programming fasion so many logical errors could be avoided on the first pass.

After the second coding session, Jacob decided to complete the rest of _insert() and NaiveDelete(). This code was then tested by both members to confirm that it works like it should. More in depth testing was still needed.

- Landon Moon
  - 2 hours - 2/1/24 - reading docs and started on Insert()
  - 2 hours - 2/8/24 - Finished Insert(). Helped polish and bugfix _insert()
- Jacob Holz
  - 2 hours - 2/1/24 - reading docs and started on _insert()
  - 2 hours - 2/8/24 - completed rough draft of _insert() index page traversal/splitting
  - 2 hours - 2/9/24 - completed rough draft of _insert() leaf insertion/splitting
  - 4 hours - 2/10/24 - completed rough draft of NaiveDelete()

Logical errors and how you handled them

Some logical errors were avoided before they cropped up through the use of pair programming, and reviewing line by line.

- Logical Error 1: index page splitting

  After looking through the provided library, we found out that there was no getCurent(rid) function in BTIndexPage. Both the getFirst(rid) and getNext(rid) are intended to be used as iterators but because we are changing the underlying data, this would be ill-advised. When splitting an index page, records from the first page needs to be removed and inserted into the second page. Without in-depth knowlege of the functions above, an iterator would give unusual results when deleting parts of the

original array. This logical error was avoided by using getNext(rid) as an accessor by setting rid to the previous slotNo. This gave us more control over which rid we were copying and deleting

There is probably a way to use getRecord(rid) instead of our implementation, but converting a custom Tuple class to a KeyDataEntry class felt overboard.

- Logical Error 2
- Logical Error 3
- (More errors if desired)