B+ Tree Design Report

In the constructor method, it will check if the index file already exist. The new index file will be created and read or just read depends on the case. Most important of all, unpin pages have been read is always a priority to avoid further troubles.

In the design of insertEntry, there are two different scenarios based on if the current root is a leaf. Thus, I added “rootIsLeaf” in btree.h to record the property. If the current root is a leaf, then the entry can be inserted in this page if it’s not full. I will split the root and locate appropriate leaf for insertion. If the current root is non leaf node, then I must locate the correct leaf node to insert. Similarly, the check if it’s full will be performed and split if necessary.

For startScan method, I first check if the inputs are valid. And then further locate the position of low integer value.

For endScan method, it will attempt to terminate the current scan first. It will throw ScanNotInitializedException if the scan is not on. It will further unpin the page if it goes through. PageNotPinnedException will be thrown otherwise.

Finally, I add createRelationMassive to test out by inserting one million tuples into the relation. And the testing result is in test4.