50.043 Database Systems and Big Data

Lab 4 - Spring 2024

Group Members

Name	Student ID
Atul Parida	1006184
Chandrasekar Akash	1006228
Jignesh Motwani	1006178

Summary of Code & Design Decisions

Exercise 1 (Search)

For Exercise 1 we implemented the *findLeafPage()* method, which takes a pid as input along with other parameters to return the corresponding page from the buffer pool. Steps were taken to ensure that the actual key value was present and exception handling was added in case the key was null. The page category was also verified to ensure only leaf and internal pages with read-only access could be searched through.

Exercise 2 (Insert)

For Exercise 2 we implemented the *splitLeafPage()* and *splitInternalPage()* methods, which were called whenever the number of elements held by a leaf node or internal node exceeded the node capacity, which required it to split such that each node was 50% full. Measures were taken to copy and push the middle key to the parent respectively, and the *updateParentPointer()* and *updateParentPointers()* helper functions proved very useful when updating pointer values to match the new node layouts. Nodes were handled through iteration as is the case in SimpleDB in general.

Exercise 3 & 4 (Delete & Transactions)

For Exercises 3 & 4 we implemented the *stealFromLeafPage(), stealFromLeftInternalPage(), stealFromRightInternalPage(), mergeLeafPages()* and *mergeInternalPages()* methods, to enable node merging when the number of values present contained in a node was less than 50% capacity, and to enable transaction management within the B+Tree. Our test cases initially passed all system tests except *BTreeTest*, but through some logic optimisation, we were able to handle *BTreeTest* and complete the database. This optimisation proved challenging but made our expertise in building scalable code more robust.

Incomplete Code Elements

All of the required code elements have been implemented as part of Lab 4. We can officially say that SimpleDB is complete.