Lab 4 Report

The BTreeFile.java file we have implemented for Lab 4 serves as the core component for managing B-Trees. It contains implementations for various operations including insertion, deletion, searching, and splitting of nodes. The purpose of this file would be for us to maintain the balance and integrity of the B-Tree structure, ensuring efficient data retrieval.

Exercise 1: Implement BTreeFile.findLeafPage()

In the findLeafPage method, design decisions were made to efficiently navigate the B-Tree structure and locate the leaf page where a key should be inserted or found. It manages to proceed with this task by recursively traversing through the internal nodes until it reaches the leaf page corresponding to the provided key value. By iterating through the entries in the internal page, it determines the appropriate child page based on the key value. These entries consist of key-value pairs and pointers to child pages. By comparing the provided key value with the keys in the internal page we can determine the appropriate child page that will proceed with the traversal. Handling cases where the provided key value is null ensures finding the left-most leaf page, essential for scanning the entire file. Using BTreeFile.getPage() for fetching pages ensures proper tracking of dirty pages, crucial for subsequent operations like insertion and deletion.

Exercise 2: Implement BTreeFile.splitLeafPage() and BTreeFile.splitInternalPage()

Splitting pages is essential for maintaining the balanced nature of the B+ tree. splitLeafPage() and splitInternalPage() handle the redistribution of tuples/entries between pages and update parent pointers accordingly. By creating new pages using getEmptyPage(), the code ensures efficient reuse of deleted pages. Utilizing iterators for leaf and internal pages facilitates iterating through tuples/entries, while functions like updateParentPointers() help maintain consistency during splits.

Exercise 3: Implement BTreeFile.stealFromLeafPage(), BTreeFile.stealFromLeftInternalPage() and BTreeFile.stealFromRightInternalPage()

Redistributing tuples/entries between pages prevents pages from becoming less than half full. These functions evenly redistribute entries between siblings or merge pages if necessary. Updating parent pointers and keys ensures the integrity of the tree structure. Utilizing BTreeFile.getPage() for fetching pages and tracking dirty pages helps maintain consistency or data integrity during redistribution operations.

Exercise 4: Implement BTreeFile.mergeLeafPages() and BTreeFile.mergeInternalPages()

Merging pages is essential for reclaiming space and maintaining the balanced nature of the B+ tree. These merging functions implemented above, perform the inverse of page splits, merging tuples or entries from two pages into one. Deleting entries from the parent and setting pages as empty ensures proper cleanup and reuse of pages. Consistent use of BTreeFile.getPage() for fetching pages and updating dirty pages ensures data integrity during merging operations even as the pages are consolidated and space is reclaimed.

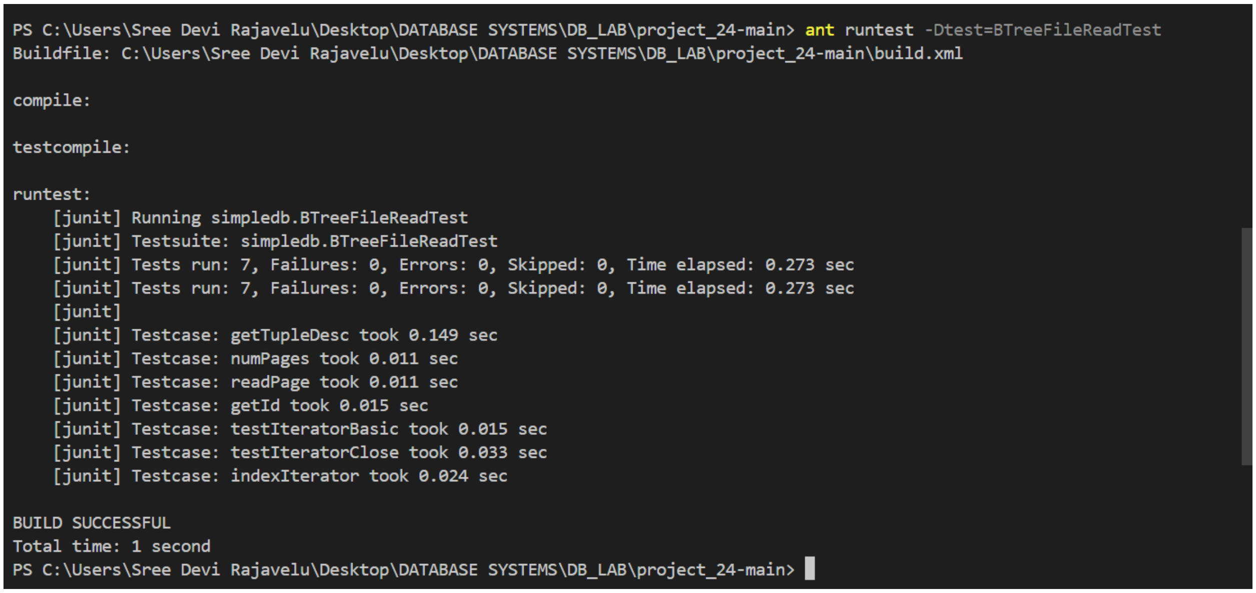
The original provided API of the Project was not changed

The code should pass all of the unit tests and system tests for the entirety of the project. Excluding the optional credit **BTreeTest** system test

Appendix

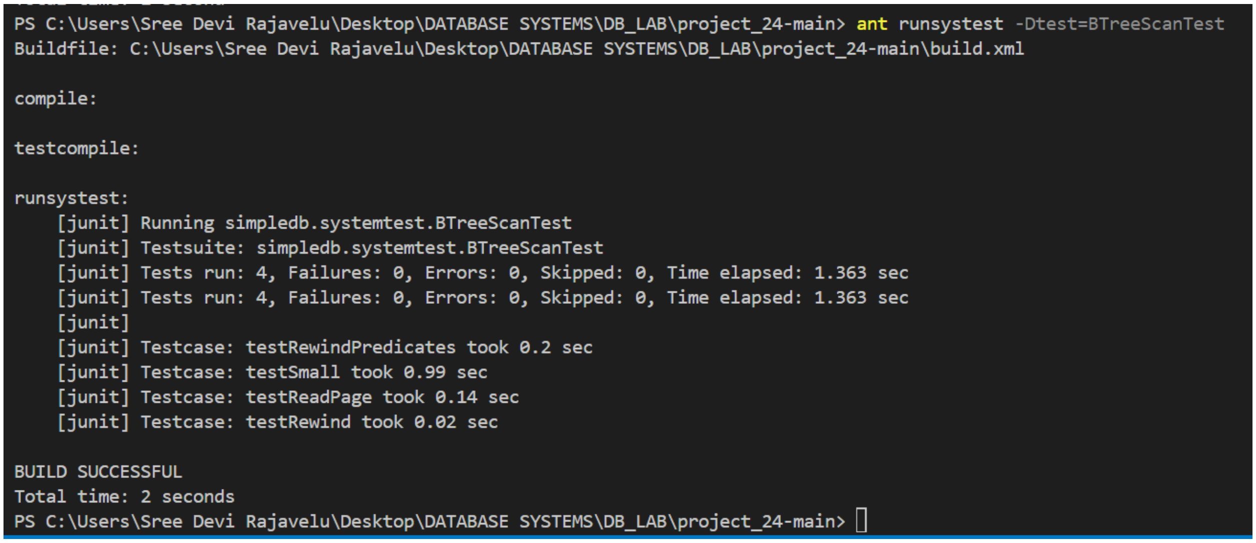
Exercise 1

Unit test: BTreeFileReadTest



```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=BTreeFileReadTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.BTreeFileReadTest
    [junit] Testsuite: simpledb.BTreeFileReadTest
    [junit] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.273 sec
    [junit] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.273 sec
    [junit]
    [junit] Testcase: getTupleDesc took 0.149 sec
    [junit] Testcase: numPages took 0.011 sec
    [junit] Testcase: readPage took 0.011 sec
    [junit] Testcase: getId took 0.015 sec
    [junit] Testcase: testIteratorBasic took 0.015 sec
    [junit] Testcase: testIteratorClose took 0.033 sec
    [junit] Testcase: indexIterator took 0.024 sec

BUILD SUCCESSFUL
Total time: 1 second
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

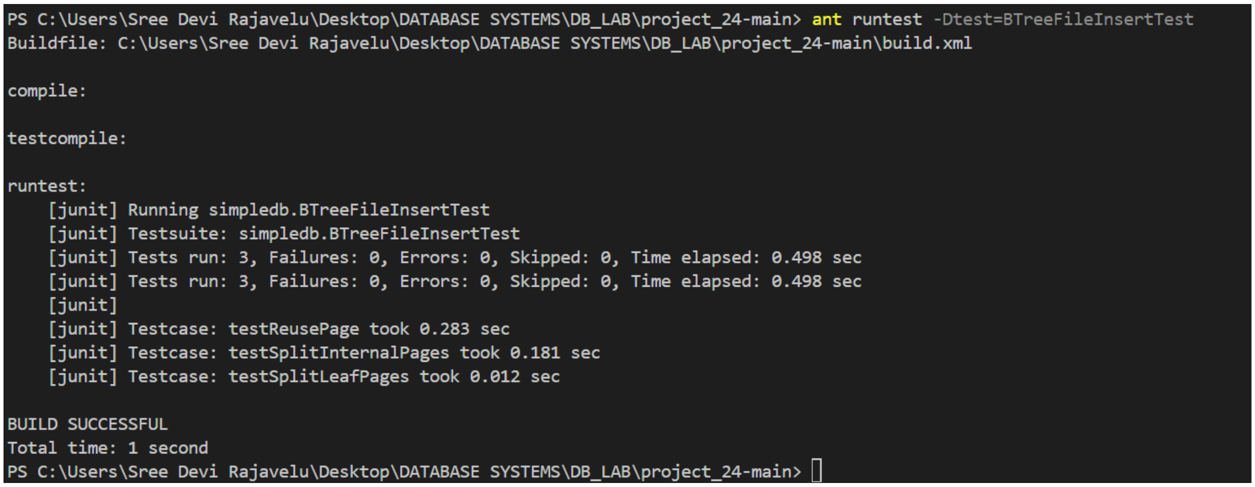
System Test: BTreeScanTest



```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runsystest -Dtest=BTreeScanTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runsystest:
    [junit] Running simpledb.systemtest.BTreeScanTest
    [junit] Testsuite: simpledb.systemtest.BTreeScanTest
    [junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.363 sec
    [junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.363 sec
    [junit]
    [junit] Testcase: testRewindPredicates took 0.2 sec
    [junit] Testcase: testSmall took 0.99 sec
    [junit] Testcase: testReadPage took 0.14 sec
    [junit] Testcase: testRewind took 0.02 sec

BUILD SUCCESSFUL
Total time: 2 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

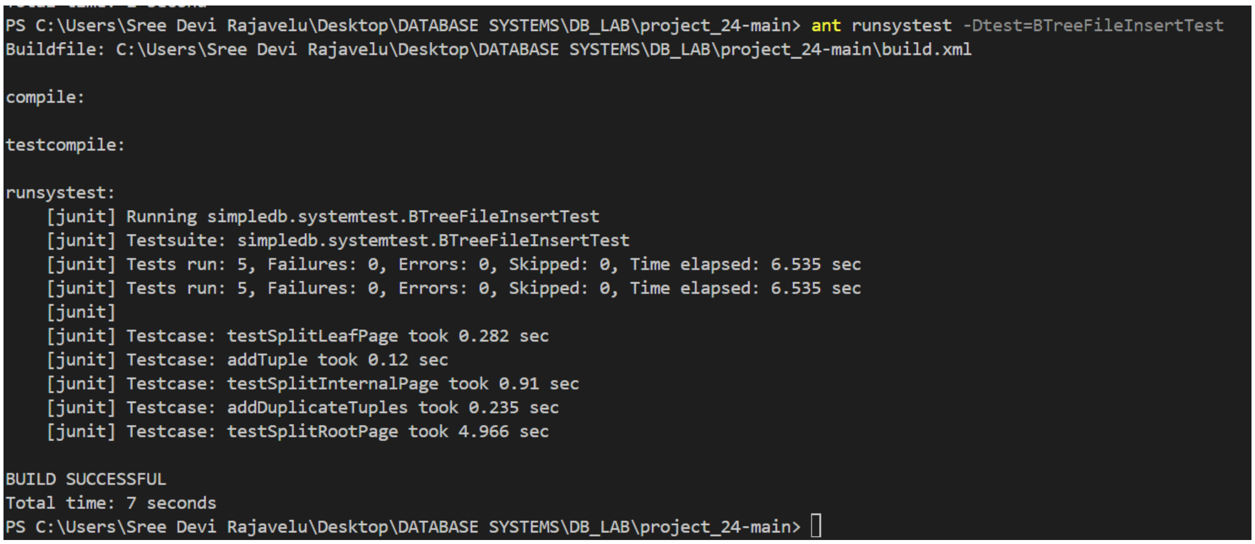
Exercise 2:

Unit test: BTreeFileInsertTest



```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=BTreeFileInsertTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.BTreeFileInsertTest
    [junit] Testsuite: simpledb.BTreeFileInsertTest
    [junit] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.498 sec
    [junit] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.498 sec
    [junit]
    [junit] Testcase: testReusePage took 0.283 sec
    [junit] Testcase: testSplitInternalPages took 0.181 sec
    [junit] Testcase: testSplitLeafPages took 0.012 sec

BUILD SUCCESSFUL
Total time: 1 second
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> 
```

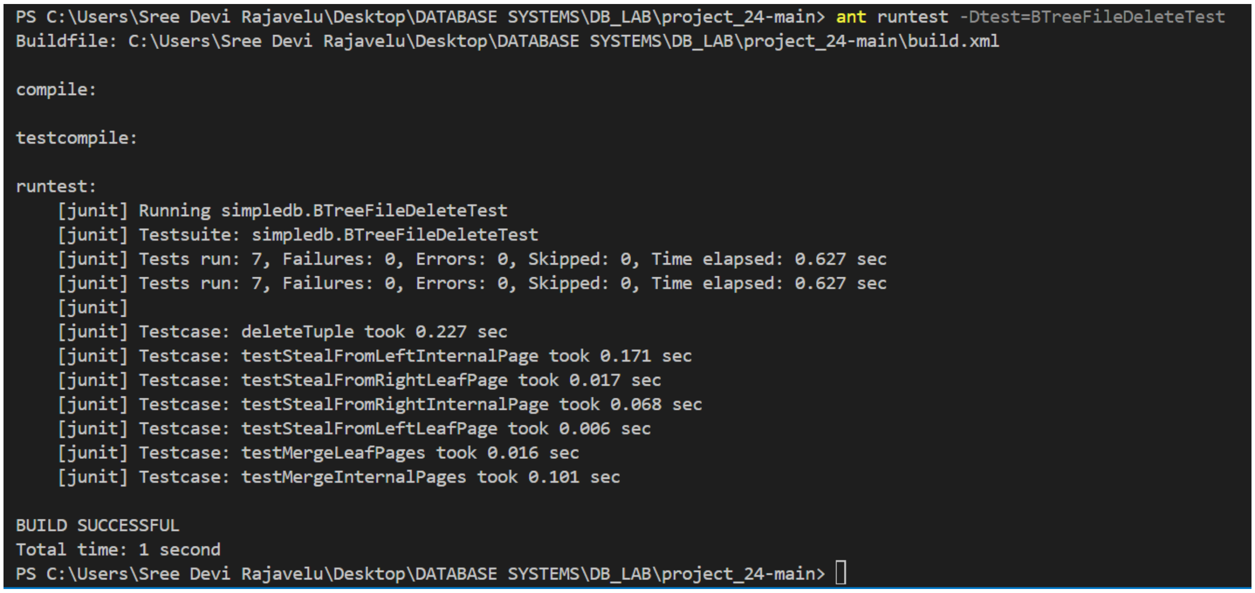
System test: BTreeFileInsertTest



```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runsystest -Dtest=BTreeFileInsertTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runsystest:
    [junit] Running simpledb.systemtest.BTreeFileInsertTest
    [junit] Testsuite: simpledb.systemtest.BTreeFileInsertTest
    [junit] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 6.535 sec
    [junit] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 6.535 sec
    [junit]
    [junit] Testcase: testSplitLeafPage took 0.282 sec
    [junit] Testcase: addTuple took 0.12 sec
    [junit] Testcase: testSplitInternalPage took 0.91 sec
    [junit] Testcase: addDuplicateTuples took 0.235 sec
    [junit] Testcase: testSplitRootPage took 4.966 sec

BUILD SUCCESSFUL
Total time: 7 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> 
```

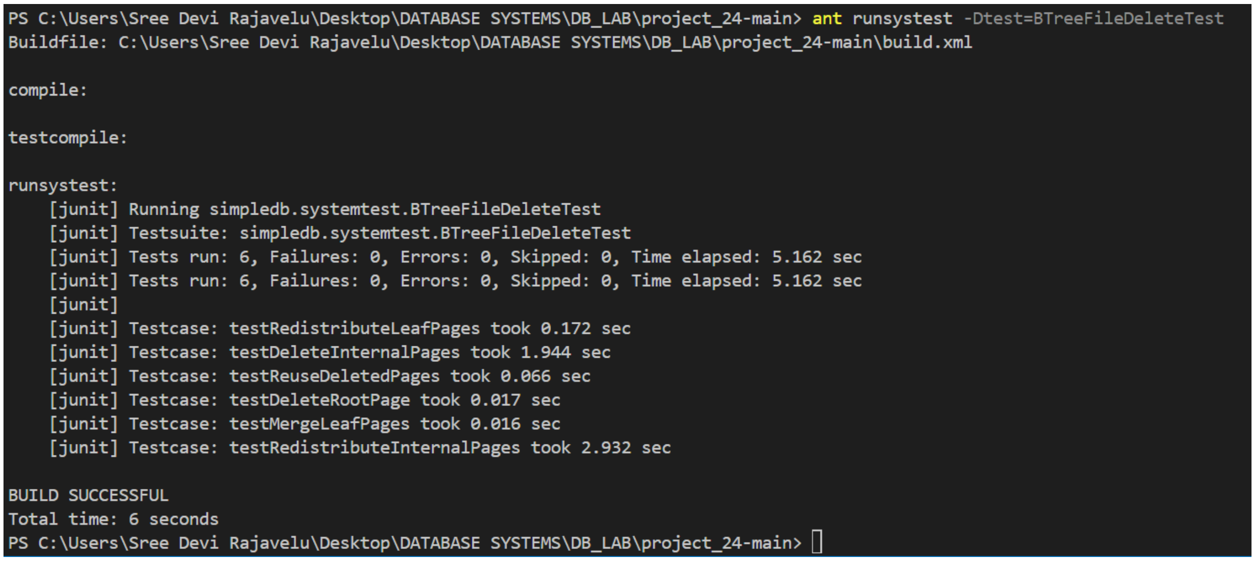
Exercise 3 & 4:

Unit test: BTreeFileDeleteTest



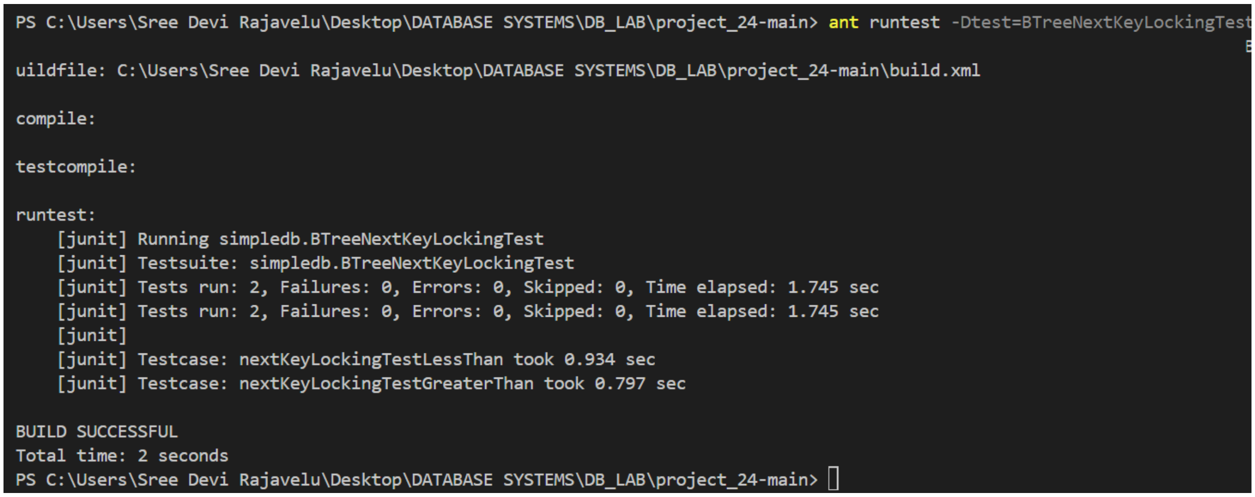
```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=BTreeFileDeleteTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.BTreeFileDeleteTest
    [junit] Testsuite: simpledb.BTreeFileDeleteTest
    [junit] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.627 sec
    [junit] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.627 sec
    [junit]
    [junit] Testcase: deleteTuple took 0.227 sec
    [junit] Testcase: testStealFromLeftInternalPage took 0.171 sec
    [junit] Testcase: testStealFromRightLeafPage took 0.017 sec
    [junit] Testcase: testStealFromRightInternalPage took 0.068 sec
    [junit] Testcase: testStealFromLeftLeafPage took 0.006 sec
    [junit] Testcase: testMergeLeafPages took 0.016 sec
    [junit] Testcase: testMergeInternalPages took 0.101 sec

BUILD SUCCESSFUL
Total time: 1 second
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> 
```

System test: BTreeFileDeleteTest



```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runsystest -Dtest=BTreeFileDeleteTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runsystest:
    [junit] Running simpledb.systemtest.BTreeFileDeleteTest
    [junit] Testsuite: simpledb.systemtest.BTreeFileDeleteTest
    [junit] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 5.162 sec
    [junit] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 5.162 sec
    [junit]
    [junit] Testcase: testRedistributeLeafPages took 0.172 sec
    [junit] Testcase: testDeleteInternalPages took 1.944 sec
    [junit] Testcase: testReuseDeletedPages took 0.066 sec
    [junit] Testcase: testDeleteRootPage took 0.017 sec
    [junit] Testcase: testMergeLeafPages took 0.016 sec
    [junit] Testcase: testRedistributeInternalPages took 2.932 sec

BUILD SUCCESSFUL
Total time: 6 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> 
```

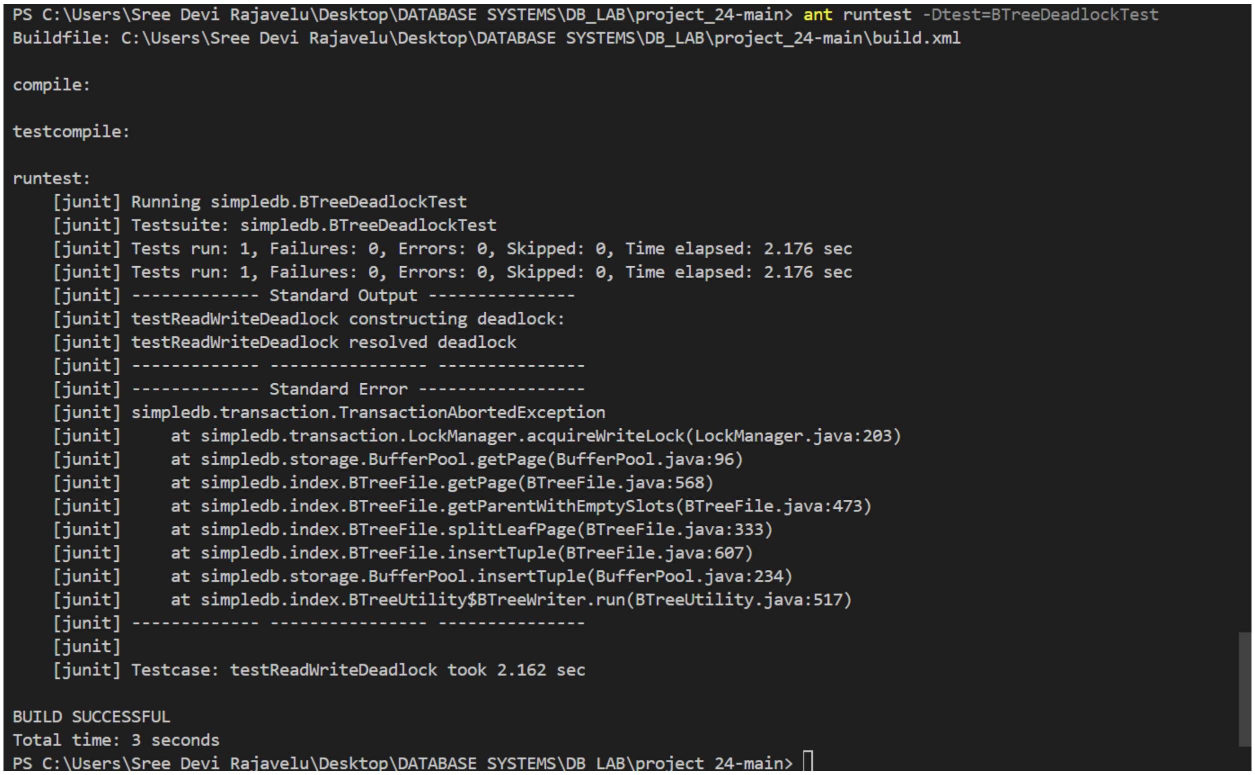
Exercise 4:
Unit test: BTreeNextKeyLockingTest



```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=BTreeNextKeyLockingTest
                                                                                                                              B
uildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.BTreeNextKeyLockingTest
    [junit] Testsuite: simpledb.BTreeNextKeyLockingTest
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.745 sec
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.745 sec
    [junit]
    [junit] Testcase: nextKeyLockingTestLessThan took 0.934 sec
    [junit] Testcase: nextKeyLockingTestGreaterThan took 0.797 sec

BUILD SUCCESSFUL
Total time: 2 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> 
```

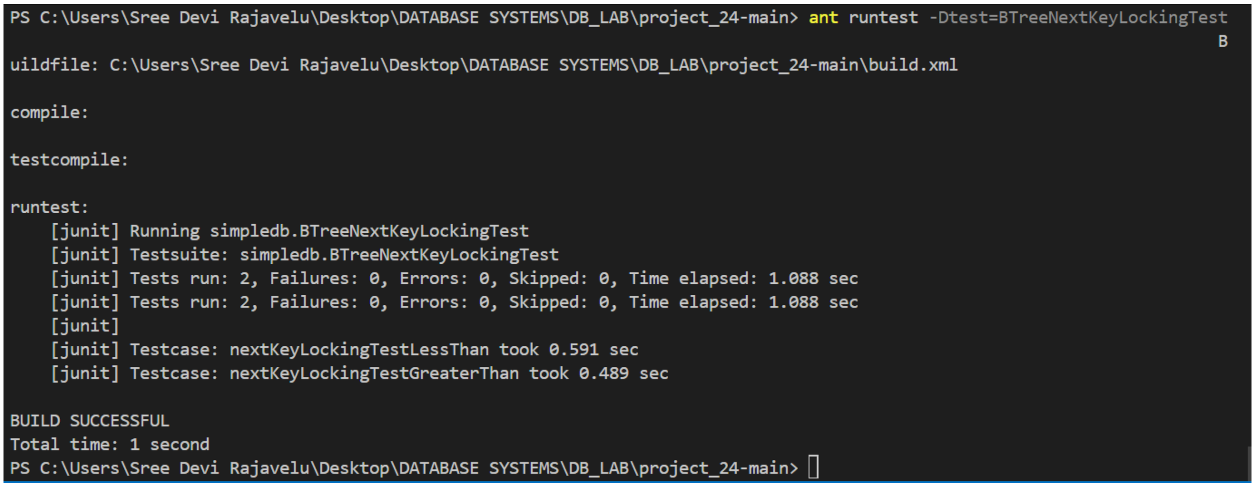
Exercise 5:

Unit test: BTreeDeadlockTest



```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=BTreeDeadlockTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.BTreeDeadlockTest
    [junit] Testsuite: simpledb.BTreeDeadlockTest
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.176 sec
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.176 sec
    [junit] ------------- Standard Output ---------------
    [junit] testReadWriteDeadlock constructing deadlock:
    [junit] testReadWriteDeadlock resolved deadlock
    [junit] ------------- ---------------- ---------------
    [junit] ------------- Standard Error -----------------
    [junit] simpledb.transaction.TransactionAbortedException
    [junit]     at simpledb.transaction.LockManager.acquireWriteLock(LockManager.java:203)
    [junit]     at simpledb.storage.BufferPool.getPage(BufferPool.java:96)
    [junit]     at simpledb.index.BTreeFile.getPage(BTreeFile.java:568)
    [junit]     at simpledb.index.BTreeFile.getParentWithEmptySlots(BTreeFile.java:473)
    [junit]     at simpledb.index.BTreeFile.splitLeafPage(BTreeFile.java:333)
    [junit]     at simpledb.index.BTreeFile.insertTuple(BTreeFile.java:607)
    [junit]     at simpledb.storage.BufferPool.insertTuple(BufferPool.java:234)
    [junit]     at simpledb.index.BTreeUtility$BTreeWriter.run(BTreeUtility.java:517)
    [junit] ------------- ---------------- ---------------
    [junit]
    [junit] Testcase: testReadWriteDeadlock took 2.162 sec

BUILD SUCCESSFUL
Total time: 3 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> 
```

Unit test: BTreeNextKeyLockingTest



```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=BTreeNextKeyLockingTest
                                                                                                                         B
uildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.BTreeNextKeyLockingTest
    [junit] Testsuite: simpledb.BTreeNextKeyLockingTest
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.088 sec
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.088 sec
    [junit]
    [junit] Testcase: nextKeyLockingTestLessThan took 0.591 sec
    [junit] Testcase: nextKeyLockingTestGreaterThan took 0.489 sec

BUILD SUCCESSFUL
Total time: 1 second
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```