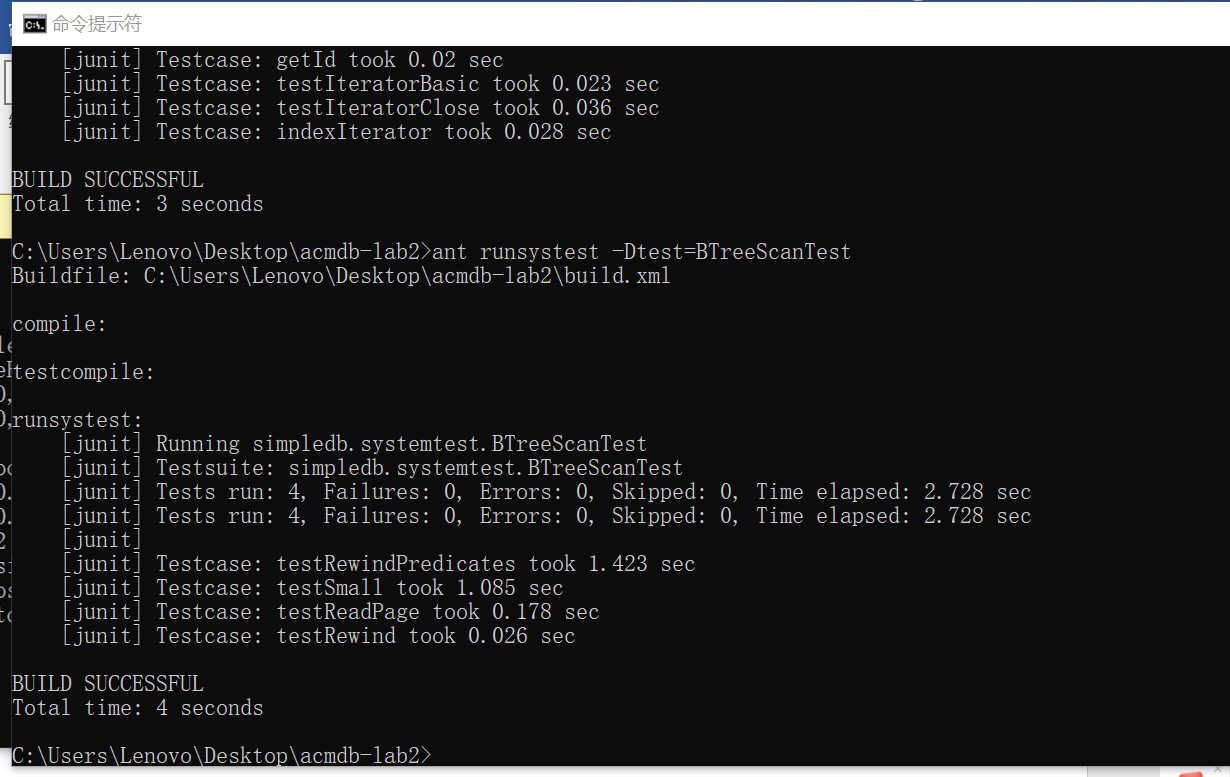
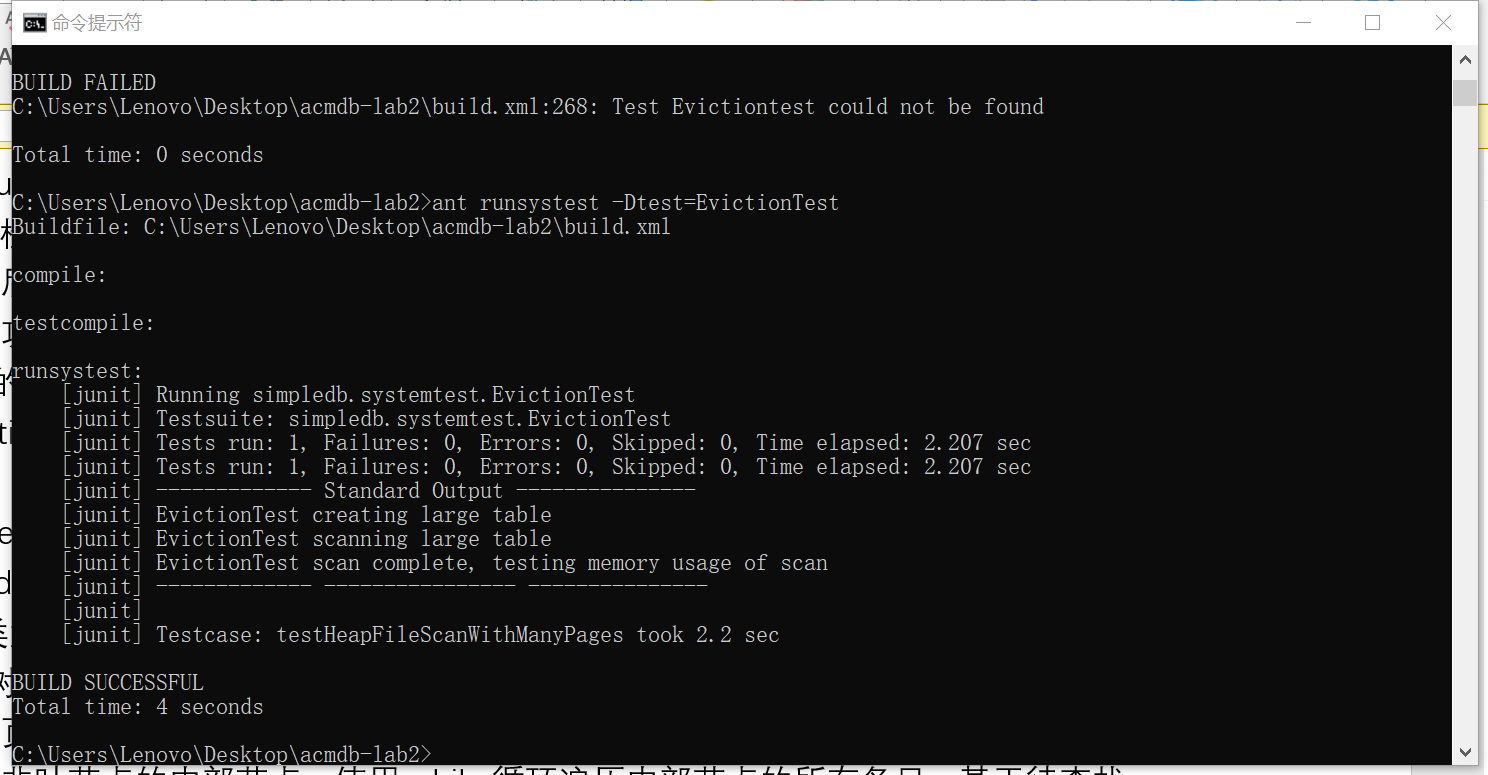
本次实验分为page evction和B+tree两个部分，在lab1的基础上我完成了readme对应java文件的编写，总用时4天（20个小时左右）。

1. Page eviction

需要在bufferpool的相关函数中实现页面淘汰。在flushpage中，记录了页面是否为“脏”的参数，检查页面“脏”状态决定是否刷写，记录日志确保事务的持久性和可恢复性，最终将修改后的页面内容写入磁盘，并更新页面状态。重新整理了src下面的java函数分为不同的功能，同时选择LRUcache和lockmanagement策略，pagecache和LRUcache等相关功能的java代码在src的子目录下面。

通过Eviction test.

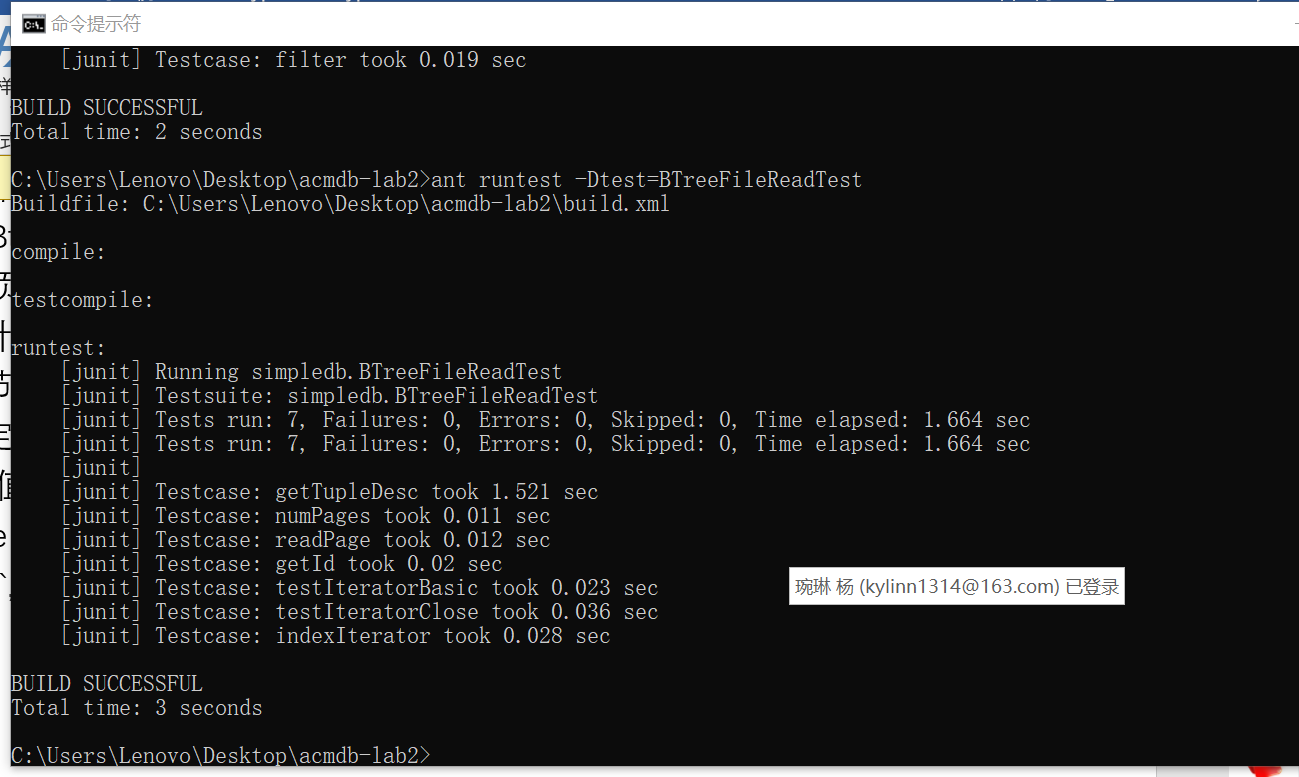
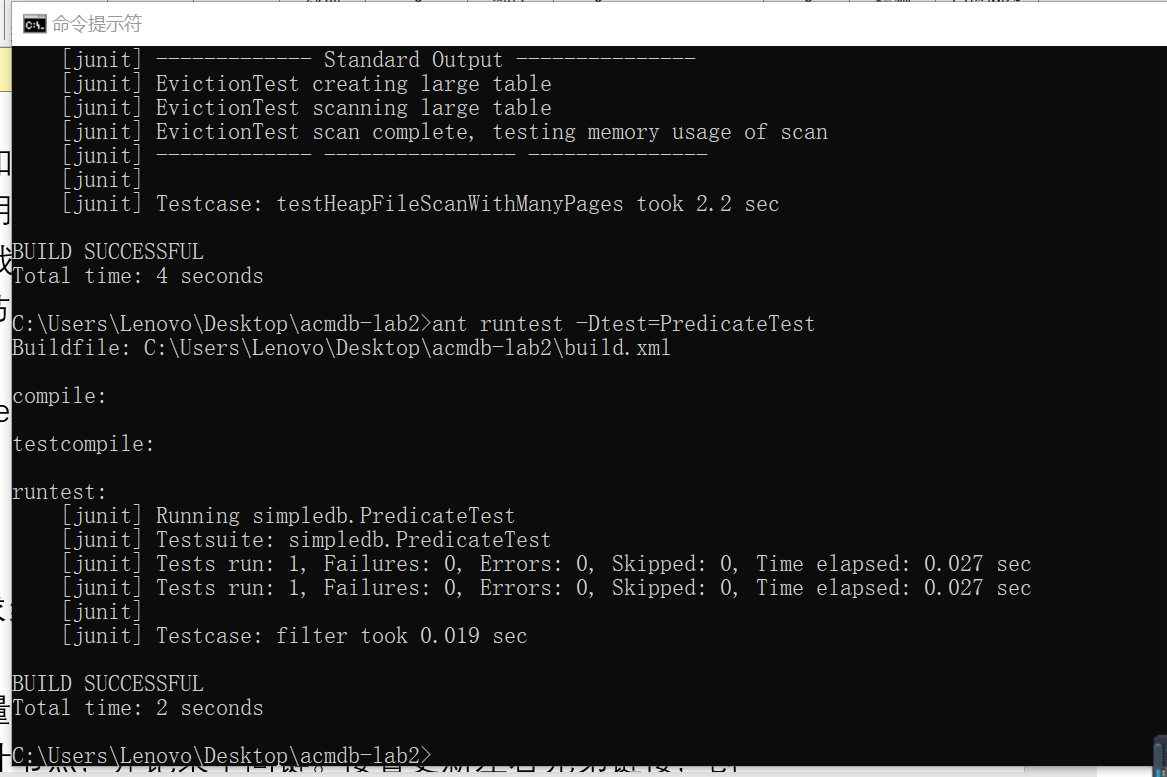


1. B+Tree Search

根据readme要求，我开始编写btreefile里的关键函数，实现递归查找包含特定字段（Field类型）的叶节点（BtreeLeafPage）

根据B+树特性，我先判断页面类别，if (type == BTreePageId.LEAF): 检查传入的页面ID所代表的页面类别是否为叶节点。如果是，直接跳过至后面。注意权限是READ\_ONLY.然后处理非叶节点的内部节点，使用while循环遍历内部节点的所有条目，基于待查找字段值与条目键的关系决定递归查找左子节点还是右子节点。通过递归遍历B树结构，最终找到包含指定字段值的叶节点并返回。同时为了实现功能，我也完成了IndexPredicate和Predicate。

通过PredicateTest.java`, `BTreeFileReadTest.java` and the system tests in `BTreeScanTest.java.



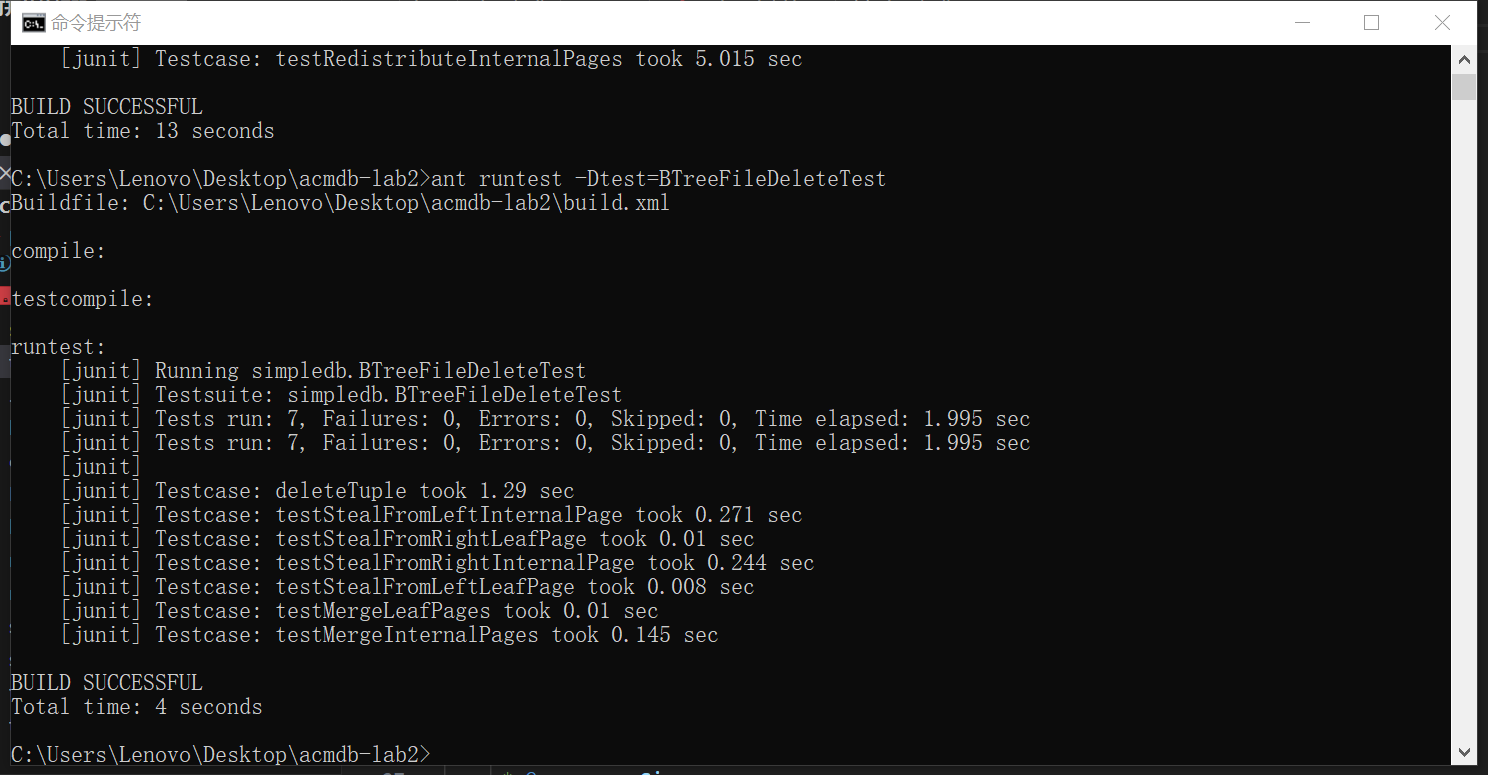
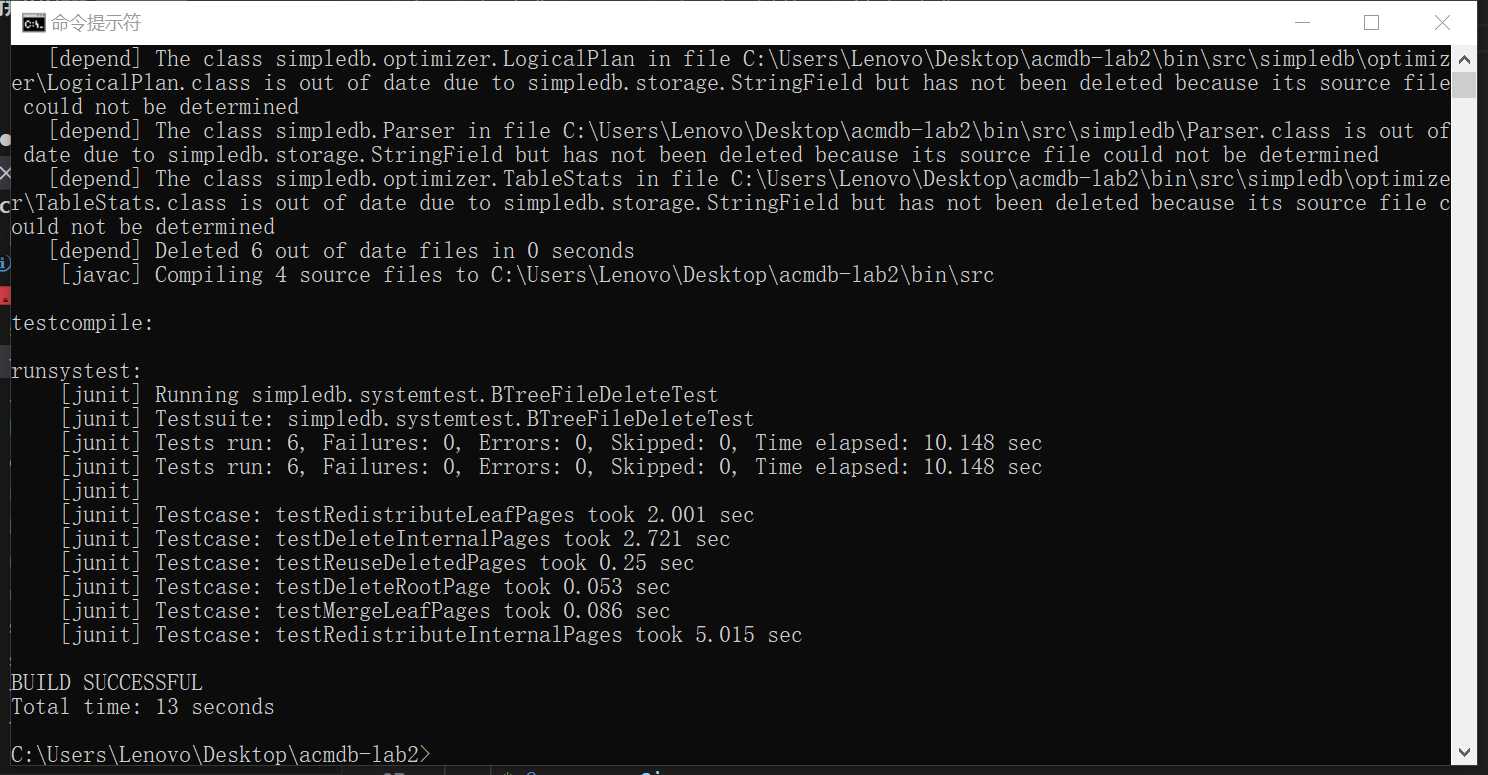
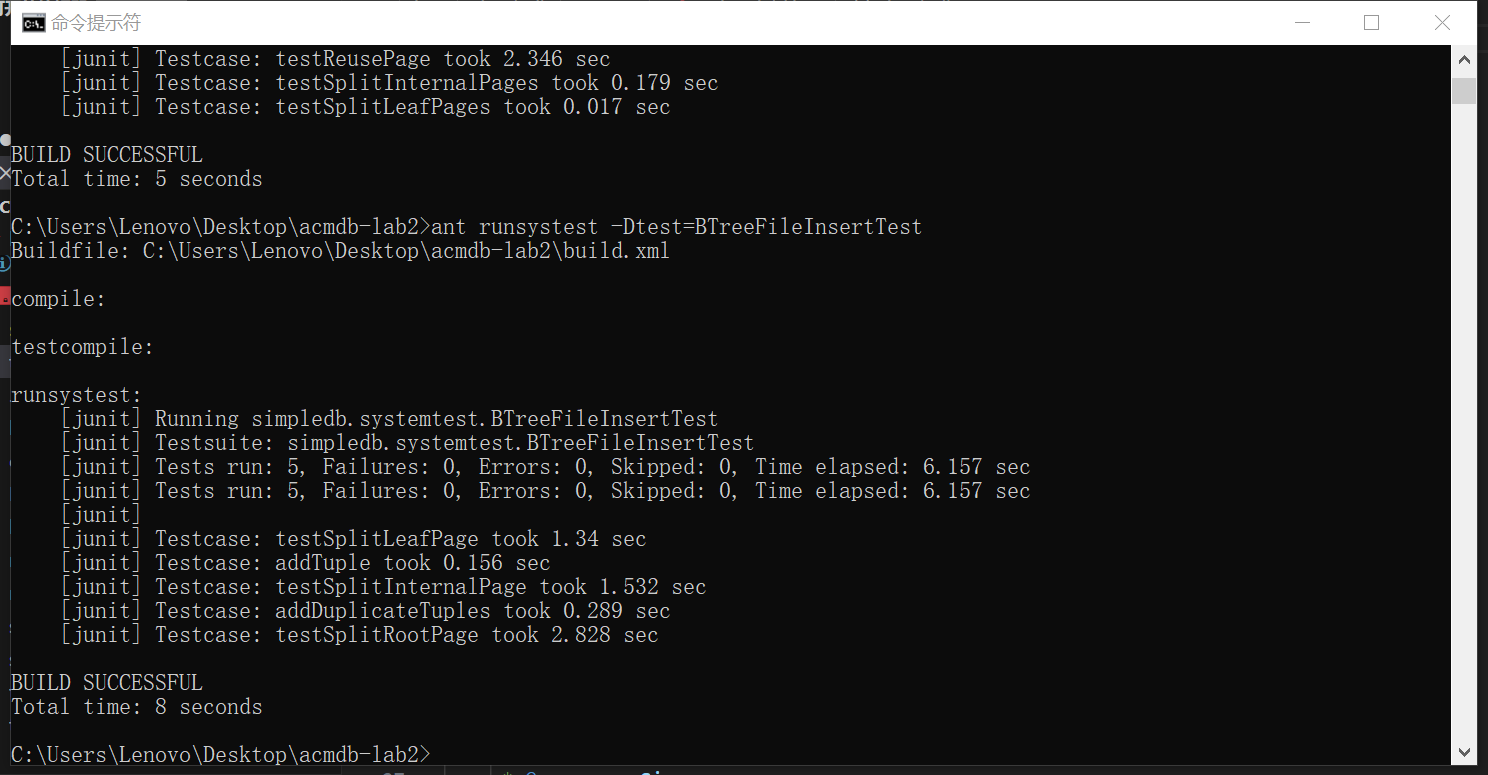
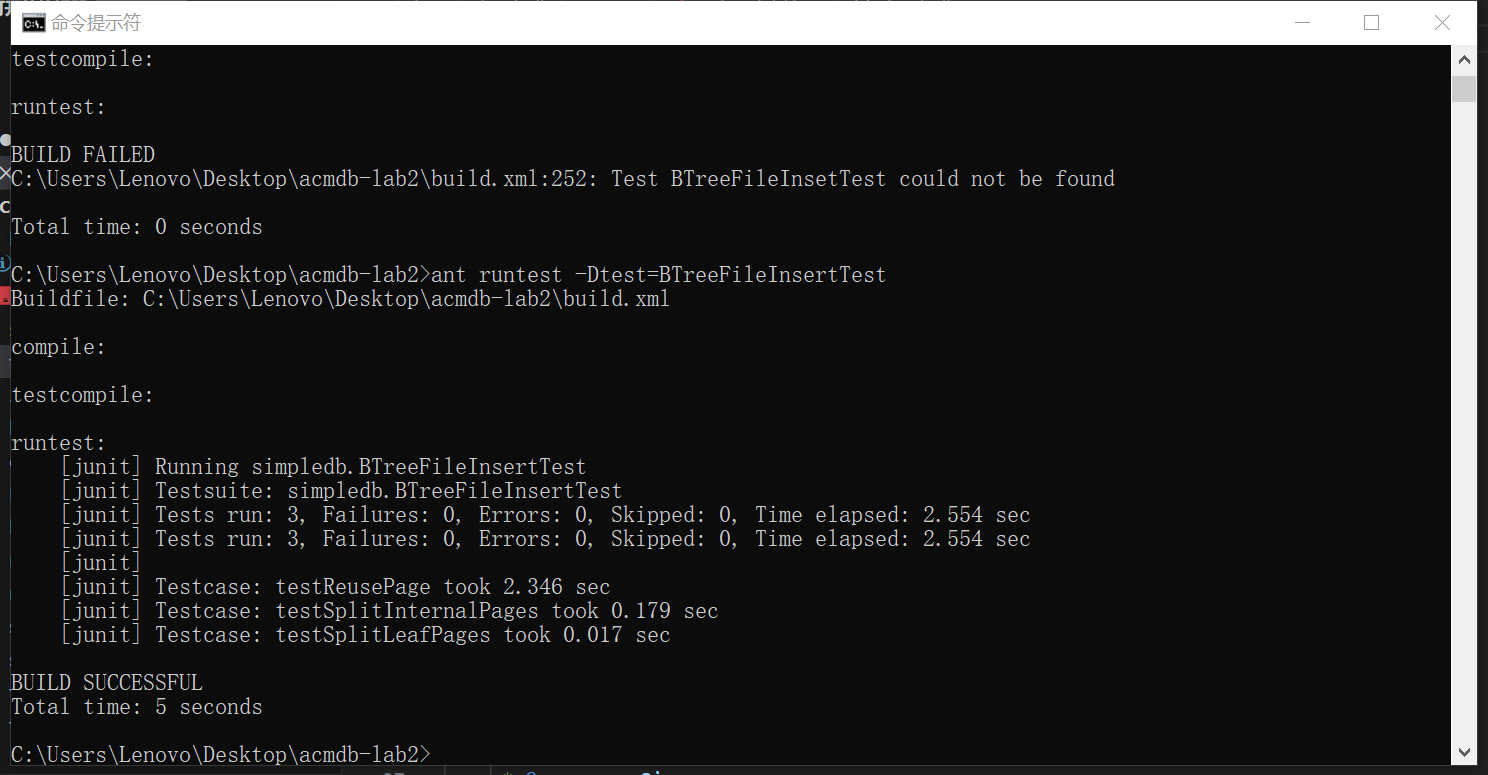
1. B+Tree insert

因为lab顺序问题，我先按照要求编写了`splitLeafPage()` and `splitInternalPage()` in `BTreeFile.java`。

对于满树叶节点，首先根据元组数量的一半确定分裂位置，创建新叶节点，然后逆序遍历原叶节点，将一半元组移动到新叶节点，并记录中间键。接着更新左右兄弟链接，创建内部节点条目并插入到父节点，同时更新叶节点的父节点ID。最后，根据待插入字段与中间键的关系返回合适的叶节点。

对于满树内部节点，首先创建新右内部节点，逆序遍历原内部节点，将一半条目移动到新右内部节点。接着处理中间键及其子节点指针，将其插入到父节点，并递归处理父节点的分裂（如果被判定需要）。然后更新脏页集合，更新所有受影响子节点的父节点指针。最后，根据待插入字段与中间键的关系返回合适的内部节点。

通过了BTreeFileInsertTest.java和系统测试



1. B+Tree delete

叶节点窃取：根据传入的参数确定是从左兄弟中“窃取”，还是从右兄弟中“窃取”。根据兄弟节点中tuple的数量，确定“窃取的数量”。参数entry是父节点中指向page和其兄弟节点的entry，将entry的key更改为page和其兄弟节点key的中间值。

内部节点窃取：根据page及其左兄弟中key的数量，确定从其做兄弟中“窃取”几个key。因为内部节点与其父节点中的key值没有重复，迁移key的时候也需要将父节点中的key移动到page中。然后将page左兄弟节点中的key平均分配。分配之后，将page左兄弟节点中最大的key“挤到”父节点中。最后进行更新。右兄弟同理。

叶节点merge:将rightPage中的所有tuple添加到leftPage中。然后判断rightPage是否有右兄弟，如果没有leftPage的右兄弟为空，如果有leftPage的右兄弟指向rightPage的右兄弟。调用setEmptyPage方法将rightPage在header标记为空。调用deleteParentEntry方法，从父级中删除左右孩子指针指向leftPage和rightPage的entry。最后将leftPage与parent添加到dirtypages中。

内部节点merge：先将父节点中的指向leftPage和rightPage的entry添加到leftPage中然后将rightPage中的entry添加到leftPage中。更新leftPage孩子节点的指针，调用setEmptyPage方法将rightPage在header标记为空。调用deleteParentEntry方法，从父级中删除左右孩子指针指向leftPage和rightPage的entry。最后将leftPage与parent添加到dirtypages中

通过`BTreeFileDeleteTest.java`.和系统测试

五．编写Haeppage和bufferpool中的insert和delete功能。如何重新进行insert和delete的系统测试