代码修改文件在 acmdb-lab2/acmdb-lab0 中。

实验结果如下：

Package Explorer JUnit

Finished after 2.269 seconds

Runs: 5/5   Errors: 0   Failures: 0

- simpledb.systemtest.BTreeFileInsertTest [Runner: J
  - addTuple (0.001 s)
  - addDuplicateTuples (0.140 s)
  - testSplitLeafPage (0.031 s)
  - testSplitRootPage (1.678 s)
  - testSplitInternalPage (0.274 s)

```java
1  package simpledb;
2  import javax.xml.crypto.Data;
7
8  /**
9   * BufferPool manages the reading and writing of pages into memory from
10  * disk. Access methods call into it to retrieve pages, and it fetches
11  * pages from the appropriate location.
12  * <p>
13  * The BufferPool is also responsible for locking;  when a transaction fetches
14  * a page, BufferPool checks that the transaction has the appropriate
15  * locks to read/write the page.
16  *
17  * @Threadsafe, all fields are final
18  */
19  public class BufferPool {
20      /** Bytes per page, including header. */
21      private static final int PAGE_SIZE = 4096;
22
23      private static int pageSize = PAGE_SIZE;
24      private final int numPages;
25      private final ConcurrentHashMap<PageId, Page> Pages;
26      /** Default number of pages passed to the constructor. This is used by
27      other classes. BufferPool should use the numPages argument to the
28      constructor instead. */
29      public static final int DEFAULT_PAGES = 50;
30
31      /**
32       * Creates a BufferPool that caches up to numPages pages
```

Package Explorer JUnit

Finished after 0.23 seconds

Runs: 3/3   Errors: 0   Failures: 0

- simpledb.BTreeFileInsertTest [Runner: JUnit 4] (0.1
  - testSplitInternalPages (0.096 s)
  - testSplitLeafPages (0.019 s)
  - testReusePage (0.006 s)

```java
1  package simpledb;
2  import javax.xml.crypto.Data;
7
8  /**
9   * BufferPool manages the reading and writing of pages into memory from
10  * disk. Access methods call into it to retrieve pages, and it fetches
11  * pages from the appropriate location.
12  * <p>
13  * The BufferPool is also responsible for locking;  when a transaction fetches
14  * a page, BufferPool checks that the transaction has the appropriate
15  * locks to read/write the page.
16  *
17  * @Threadsafe, all fields are final
18  */
19  public class BufferPool {
20      /** Bytes per page, including header. */
21      private static final int PAGE_SIZE = 4096;
22
23      private static int pageSize = PAGE_SIZE;
24      private final int numPages;
25      private final ConcurrentHashMap<PageId, Page> Pages;
26      /** Default number of pages passed to the constructor. This is used by
27      other classes. BufferPool should use the numPages argument to the
28      constructor instead. */
29      public static final int DEFAULT_PAGES = 50;
30
```

Package Explorer JUnit

Finished after 2.16 seconds

Runs: 6/6   Errors: 0   Failures: 0

- simpledb.systemtest.BTreeFileDeleteTest [Runner:
  - testMergeLeafPages (0.000 s)
  - testDeleteRootPage (0.000 s)
  - testRedistributeLeafPages (0.000 s)
  - testRedistributeInternalPages (1.577 s)
  - testReuseDeletedPages (0.033 s)
  - testDeleteInternalPages (0.314 s)

Failure Trace

```java
1  package simpledb;
2  import javax.xml.crypto.Data;
7
8  /**
9   * BufferPool manages the reading and writing of pages into memory from
10  * disk. Access methods call into it to retrieve pages, and it fetches
11  * pages from the appropriate location.
12  * <p>
13  * The BufferPool is also responsible for locking;  when a transaction fetches
14  * a page, BufferPool checks that the transaction has the appropriate
15  * locks to read/write the page.
16  *
17  * @Threadsafe, all fields are final
18  */
19  public class BufferPool {
20      /** Bytes per page, including header. */
21      private static final int PAGE_SIZE = 4096;
22
23      private static int pageSize = PAGE_SIZE;
24      private final int numPages;
25      private final ConcurrentHashMap<PageId, Page> Pages;
26      /** Default number of pages passed to the constructor. This is used by
27      other classes. BufferPool should use the numPages argument to the
28      constructor instead. */
29      public static final int DEFAULT_PAGES = 50;
30
31      /**
32       * Creates a BufferPool that caches up to numPages pages.
33       *
34       * @param numPages maximum number of pages in this buffer pool.
35       */
36      public BufferPool(int numPages) {
37          // some code goes here
38          this.numPages=numPages;
```

Package Explorer JUnit

Finished after 0.402 seconds

Runs: 7/7   Errors: 0   Failures: 0

- simpledb.BTreeFileDeleteTest [Runner: JUnit 4] (0.
  - deleteTuple (0.006 s)
  - testMergeLeafPages (0.014 s)
  - testStealFromLeftLeafPage (0.010 s)
  - testStealFromRightLeafPage (0.005 s)
  - testStealFromLeftInternalPage (0.096 s)
  - testStealFromRightInternalPage (0.079 s)
  - testMergeInternalPages (0.105 s)

Failure Trace

```java
1  package simpledb;
2  import javax.xml.crypto.Data;
7
8  /**
9   * BufferPool manages the reading and writing of pages into memory from
10  * disk. Access methods call into it to retrieve pages, and it fetches
11  * pages from the appropriate location.
12  * <p>
13  * The BufferPool is also responsible for locking;  when a transaction fetches
14  * a page, BufferPool checks that the transaction has the appropriate
15  * locks to read/write the page.
16  *
17  * @Threadsafe, all fields are final
18  */
19  public class BufferPool {
20      /** Bytes per page, including header. */
21      private static final int PAGE_SIZE = 4096;
22
23      private static int pageSize = PAGE_SIZE;
24      private final int numPages;
25      private final ConcurrentHashMap<PageId, Page> Pages;
26      /** Default number of pages passed to the constructor. This is used by
27      other classes. BufferPool should use the numPages argument to the
28      constructor instead. */
29      public static final int DEFAULT_PAGES = 50;
30
31      /**
32       * Creates a BufferPool that caches up to numPages pages.
33       *
34       * @param numPages maximum number of pages in this buffer pool.
```