CSE 510: PROJECT PHASE - I

Group Members (Group 16)

Anjana Ouseph
Lowkya Vuppu
Sai Anurag Vellanki
Nagarjun Reddy Mora
Chandra Sai Chalicheemala

Submitted By

Anjana Ouseph
ASU ID (emplid):1225797497
ASURITE User ID:aouseph

Abstract

Minibase is a single-user database management system, primarily intended for educational purposes, allowing students to study and implement different DBMS components without the support for concurrency control and recovery. The system includes a parser, optimizer, buffer pool manager, storage methods (such as heap files and secondary B+ tree-based indexes), and a disk space management system. The aim of Minibase is to offer students a functional DBMS where they can study and implement individual components.

Minibase has a graphical user interface that supports a subset of QBE and visualization tools for buffer management, B+ trees, query optimization, and database design, including creating ER diagrams and normalization. Raghu Ramakrishnan developed it in collaboration with the text "Database Management Systems". The HTML documentation for Minibase provides a comprehensive overview of the system architecture and component interfaces, while the text offers background information on file and record organizations, buffer management, access methods, relational query languages, operator implementation techniques, query optimization, database design, concurrency control, and recovery. All the tests have pre-defined static inputs except for the BTree Test, which needs user interaction.

Keywords: Minibase, BTree, heap files, buffer manager, database.

Introduction

In phase-1 of the project, we are supposed to set up Minibase in our local Unix system and run the tests and gain an understanding of each of the tests. In this report, I am going to briefly explain every test and its subtests briefly: its inputs, output, and other functionalities. All the tests have predetermined inputs except the BTree tests, where the user is supposed to interact with the test.

Description of Tests

BM Test

The BMTest.class's main method is executed, which leads to the creation of a BMDriver object. The BMDriver, which is a subclass of TestDriver, provides methods for testing the buffer manager. These tests are initiated by calling the runTests() method on the BMDriver object. The runTests() method first terminates any lingering processes from the previous run, then carries out six tests. If all of the tests are successful, the program terminates with a status code of 0, indicating successful execution. If any of the tests fail, a non-zero status code will be returned.

Test 1:
Overrides test1 function of the TestDriver class. This test tests some normal buffer manager operations.

We make sure at least one page has to be written during the test with the help of numPages variable.

We create the first page and then unpin the page to simplify the loop. If there is any error in each step, they are handled with the help of a try-catch block (Exception handling).

Next, we write something on each page by using for loop to iterate over each page ids and pin the page ids. After successfully pinning the pages one by one, we copy page no+ 99999 (data) onto each page. After this is successfully done we unpin the pages.

Using for loop, we iterate again over each page ids and pin each page ids, then we try to get the data value (page no + 99999) from each of the page ids. If the data values are successfully read from each page ids then we again unpins each page.

After the above steps are successfully completed we free the pages.

Test 2:
Overrides test2 function of the TestDriver class. This tests whether illegal operations can be caught.

Case 1:
We get the no. of unpinned buffers. Then we try to pin more pages than there are frames, First, allocate a page and then pin enough pages iteratively so that there is no more room and we make sure the buffer manager thinks there is no more room. Now we try to pin one more page and it fails as expected.

Case 2:
We try to pin the first page again. Then we try to free a double-pinned page. It fails.
As per the code, It's not possible to free a doubly pinned page.

Case 3:
We try to unpin a page, not in the buffer pool. Here we are trying to unpin the last page (which we tried to pin in case 1 but we couldn't) which was actually never pinned in case 1. So it fails as we are trying to unpin a page which was not pinned before.

After this, we iteratively free all the pinned pages from the buffer.

Test 3:
Overrides test3 function of the TestDriver class. It exercises some of the internal of the buffer manager.
We try to allocate and dirty some new pages one at a time and we leave some of them pinned. Copy the page number + 99999 onto each page (data). Then we randomly pin some pages as per the criteria if the page id modulus 20 equals 12. (Criteria based on a bug report). Then we read these pages back in the same order and check if they have the correct data value. While unpinning the pages that we just read, we would again unpin the page if the page has the page id % 20 equals 12.

DB TEST

The DBTest.class's main method is executed, resulting in the creation of a DBDriver object. The DBDriver, which is a subclass of TestDriver, offers functions for testing disk space management. This is activated by calling the runTests() method on the DBDriver object. Before executing the tests, the runTests() method terminates any lingering processes from the previous run, then performs six tests before starting the cleanup process.

Test 1

Test 1 creates a new database and does some tests of normal operations. We add six file entries to the database. We try to allocate a run of pages. Then we try to write something on some of them. Then we deallocate the rest of the pages.

Test 2

Test 2 opens the database created in test 1 and does some further tests.
Delete the first three file entries. Then we look up file entries that should still be there, i.e. the remaining three file entries. Then we read back from the 20 pages we wrote in test1. If there is any error, we throw the exception "Data read does not match what was written on the page".
After test2 is successful, we do a final cleanup using JavabaseDB.deallocate_page for 20 pages.

Test 3

Test 3 tests for some error conditions.

a) First, try to look up a deleted file entry (file1 which was deleted in test2). This is supposed to fail as expected and pgid returned should be null.
b) Next, try to delete a deleted page again. This is supposed to fail as expected because we are deleting an already deleted page.
c) We try to delete a non-existent file entry. This should also fail as expected.
d) Look up a file entry which is nonexistent. This should also fail.
e) Try to add a file entry that's already present. This should fail as expected with a duplicate entry exception.
f) We add a file whose name is too long. It should fail because of the long name with FileNameTooLongException.
g) Try to allocate with a run of pages that is too long. It should fail as expected.
h) Try to allocate negative run of pages. This should fail as we are allocating negative run of pages.
i) Try to deallocate negative run of pages. This should fail as we are deallocating negative run of pages.

Test 4

Test 4 tests for boundary conditions and are specific to the implementation.
Create a new database that is big enough to hold 2 pages. Make sure no pages are pinned in the database.

a) Allocate all pages remaining after DB overhead is accounted for, i.e. MAX_SPACE * 8+1-3, which is the maximum possible.
   We check if the first page is allocated to be page 3. Then checks if no. of unpinned pages is equal to the number of buffers.
   We try to allocate one more page and it fails as expected.
b) Free some of the allocated pages. We try to free pages from 3 to 9.
c) Next we free pages 33 to 40.

d) Allocate some of the just freed pages, which are adding 33 to 40.
e) We free two continued runs of allocated pages which are from 11 to 17.
f) Next, deallocate pages 18 to 28.
g) Allocate pages equaling to no. of deallocated pages, i.e. allocate pages 11 to 28.
h) Next, delete some leftover file entries which are files from 3 to 5.
i) To determine the maximum number of file entries that can fit on a single directory page, we divide MAX_SPACE by MAX_NAME. We then insert one more file entry than this maximum amount. The files are named "file + i," with "i" being the index in the loop. Despite this, after the operation, there should only be 6 pages remaining. However, we attempt to allocate 7 more pages, which results in an OutOfSpaceException. We try to allocate 6 more pages and this will pass.
j) At this point, all pages are claimed and if we try to allocate more, it should fail as expected.
k) Finally, we free the last two pages to test the boundary conditions on the space map.

Heap File Test

The HFTest.class's main method is being executed, which results in the creation of a HFDriver object. The HFDriver, which inherits from the TestDriver class, provides methods for testing the heap file. These tests are triggered by calling the runTests() method on the HFDriver object. Before starting the tests, the runTests() method terminates any processes from previous runs, then carries out six tests before initiating the cleanup process.

Test 1
Insert and scan fixed-size records. Create a heap file. Add 100 records to the file. Next, scan the records inserted. A sequential scan won't be in the same order as the insertions. However, we're inserting fixed-length records here, and in this case, the scan must return the insertion order. Each record/tuple is then scanned in the same order and we check that they have the same contents we inserted.

Test 2
In this test, we try to delete fixed-sized records. We open the same heap file used in Test 1 and we delete half the records by scanning all the records one by one and deleting odd-numbered records. We run the scan again on the remaining records and confirm that they contain their original indexes.

Test 3
Here we update fixed-sized records. Open the heap file as above test cases. Scan the records and update their rec.fvalue to 7*i (index). Then we re-scan the records to verify the updates are actually present there. Odd index records are not present here as they are deleted in test 2.

Test 4
a) Tests some error conditions. Open the same heap file as above. Try to change the size of the record to test if changing the size of the records will cause any issues. We scan

the first record and update the length to length-1 and this fails as expected, as we are shortening a record.

b) Then we try to update the length to length+1. Here also the test case fails as expected.
c) Next, we try to insert a record that's too long (4 bytes longer than minibase page size). It fails with heap.SpaceNotAvailableException exception.

BTree Test

This is the only interactive test. The execution of the BTTest.class's main method leads to the creation of a BTDriver object, which extends the TestDriver class and provides functions to test binary tree operations. These tests are initiated by calling the runTests() method on the BTDriver object. The runTests() method first terminates any lingering processes, then creates a new BTree file and presents the user with 20 options ranging from 0 to 19.

Option 0
Naive delete
A BTree is created with the name "AAA+postfix". This file has deleteFashion=0 (naive delete). For this type of deletion, we do not merge or redistribute keys.

Option 1
Full Delete
This is the default delete mechanism. Here deleteFashion=1and merging and redistribution of keys occur.

Option 2
Print the B+ Tree Structure
Prints the overall tree structure in level-wise order. Pages are printed in increasing order of their keys and levels.

Option 3
Print all Leaf Pages
Prints the contents of the leaf pages with their keys in increasing order. Each leaf page in the structure has links, referred to as left and right links, to the page preceding and following it, respectively. The initial page in the sequence has a left link of -1, and the final page has a right link of -1. The leaf pages are arranged in a doubly linked-list fashion.

Option 4
Choose a page to print.
Input the page no. which we want to be printed. It can print the root page, leaf page and internal page contents.

Option 5
Insert a Record

Inserts a new record to the BTree which is open. Inserts a single key into BTree.

Option 6

Delete a record

Deletes a record from the BTree that is open based on the key we input. It deletes a single key from BTree if it is present.

Option 7

Test1 (new file): insert n records in order

We enter the set of keys to be inserted. Here the current file is closed and it creates a new BTree and inserts records in sequential order.

Option 8

Test2 (new file): insert n records in reverse order

We enter the set of keys to be inserted. Here the current file is closed and a new BTree is created. Records are inserted in reverse fashion.

Option 9

Test3 (new file): insert n records in random order

We enter the set of keys to be inserted. Here the current file is closed and a new BTree is created. It creates an array of integers in random order but contains all the keys we insert. Then it inserts records into the BTree in the same order as the randomized array.

Option 10

Test4 (new file): insert n records in random order and delete m records randomly

We enter the ‘n’ keys to be inserted into BTree and the ‘m’ keys to be deleted randomly from what we are going to insert. The current file is closed and a new BTree is created. It creates an array of integers in random order but contains all the keys we insert. Then it inserts records into the BTree in the same order as the randomized array (like Option 9). Then the array is shuffled again and m records are deleted.

Option 11

Delete some records

Here we are asked to input lower-key and higher-key values. Based on the input we enter, the records in that range are deleted.

Option 12

Initialize a Scan

Here we are asked to input lower-key and higher-key values. Based on the input we enter, the records in that range are scanned. Please note here -3 is considered NULL for lower-key and -2 is NULL for higher-key.

Option 13

Scan the next record.

This gets the next record in the scan as option 12 and prints it. If there are no records then it throws an exception.

Option 14
Delete the just-scanned record
This deletes the record which we scanned in option 13. If no scan was opened, fails with an exception.

Option 15
Test5 (new file): insert n records in random order and delete m records randomly.
Here we are prompted to enter the no. of keys to insert and no. of keys to delete. Then it closes the current file and opens a new BTree file. It creates an array of integers in random order but contains all the keys we insert. Then it inserts records into the BTree in the same order as the randomized array. Then these randomized keys are inserted into the BTreeFile after prefixing them with "**".
This array is then again shuffled and we delete m records from it.

Option 16
Close the file
This option allows us to close the BTree that is currently open and in use.

Option 17
Open which file (input an integer for the file name)
Enter a number so that it opens an existing file. If the no. we enter is 14 it will open "AAA14". If we enter a file that is not there, it fails with an exception.

Option 18
Destroy which file (input an integer for the file name)
Enter a number so that it destroys an existing file. If the no. we enter is 14 it will destroy "AAA14". If we enter a file that is not there, it fails with an exception. When the file is destroyed all the pages are recursively freed and the file entry is deleted.

Option 19
Quit!
Quit the interactive test mode.

Index Test

The IndexTest.class's main method is being executed, which results in the creation of an IndexDriver object. The IndexDriver, a subclass of TestDriver, is responsible for running index tests. This is activated by calling the runTests() method on the IndexDriver object. The test uses two arrays, data1 (an unsorted array of strings) and data2 (a sorted version of data1), as its basis.

Two static strings data1 and data2 are defined which looks like it contains the names of peoples. Basically, both arrays contain the same names but data2 is sorted version of data1

Test 1
Create a tuple of the appropriate size to hold data1 and data2. Then a heap file is created. Then we create a BTree file called BTreeIndex. We scan the tuples from the heap file, extract the second field and use it as a key. We then insert this key vs the row id of the tuple in the heap file to the BTreeIndex. Next, we create a scan for BTreeIndex on the data of the heap file. We read tuples from the scan one by one and compare the first field in the index tuple with the list data2.

Test 2
Open the heap file and index file created in the above test. Then create an object array that matches the key with “dsilva”. Then start an index scan. If only one record has key value “dsilva”, is returned. Then the test passes.
Next, a new array is created to hold range parameters. The first object is equivalent to “all tuples greater than equals to ‘dsilva’”, while the second object is equivalent to “all tuples lesser than equals to ‘yuc’”. Combinedly both the operators are supposed to return all tuples between “dsilva” and “yuc”.
Now another IndexScan is started. We read all the records returned by the scan and compare them against “data2”. The records are returned in the same order as in data2.

Test 3
A new heap file is created and inserts 1000 records, each one with data as data[i % NUM_RECORDS], “”, random() % 1000 (this 3rd attribute is the key) and random(), where i is the index to the ith inserted record. A new index file is created. A scan is started for the tuples in the heap file and we insert into the index file a new tuple with 3 attributes and the row id of the scanned tuple.
Next, another array is created with 3 operators. The first one is “all integers greater than equal to 100” and the second one is “all integers lesser than equal to 900”. We perform an index scan on both the heap file and index file created, examining each of the output records to ensure that the key value of the current record is always equal to or greater than the key value of the previous record.

Join Test

The execution of the JoinTest.class's main method leads to the creation of a JoinDriver object, which is a subclass of TestDriver. This object is used to run join tests and is activated by calling the runTests() method on the JoinDriver object.

For this three tables are required
Sailors (sid, sname, rating, age)
Boats (bid, name, color)

Reserves (sid, bid, data)
In the JoinDriver class, we construct each of the tables (namely sailors, boats, and reserves) and insert test data in those heap files.

We have created 25 sailors, 5 boats and 10 reserves and relations between them for testing purposes.

Query1 ()
Select S.sname, R.date from Sailors as S join Reserves as R on S.sid = R.sid where R.bid = 1;

Here we are supposed to find the sailors who have boat no.1 reserved and the date of reservation.
Select S.sname, R.date from Sailors as S join Reserves as R on S.sid = R.sid where R.bid = 1;

Query2 ()
Select S.sname from Sailors as S join Reserves as R on S.sid = R.sid join Boats as B on R.bid = B.bid where B.color = ‘red’ order by S.sname;

We try to find the sailors who have boats of red color reserved and print them in lexicographical order of names.

Query3 ()
Select S.sname from Sailors as S inner join Reserves as R on S.sid = R.sid;

We are trying to find sailors who have reserved at least a boat.

Query4 ()
Select distinct S.sname from Sailors as S inner join Reserves as R on S.sid = R.sid;

Try to find the sailors who have reserved at least a boat and print their names exactly once.

Query5 ()
Select S.sname, S.rating, S.age from Sailors as S inner join Reserves as R on S.sid = R.sid where S.age > 40 || S.rating < 7;

Find sailors who are over 40 years old or have a rating below 7, who have also reserved a boat, and display their names, ratings, and ages.

Query6 ()
Select S.sname from Sailors as S inner join Reserves as R on S.sid = R.sid inner join Boats as B on B.bid = R.bid where S.rating > 7 && B.color = ‘red order by S.sname;

Find the sailors with a rating greater than 7 who have reserved a red-colored boat and print their names.

Sort Tests

The SortTest.class's main method is being executed, leading to the creation of a SORTDriver object, a subclass of TestDriver. This object is used to run sort tests, which are initiated by calling the runTests() method on the SORTDriver object. The runTests() method first terminates any lingering processes and then starts executing the tests. Six tests are run before the cleanup process begins.

Two static strings data1 and data2 are defined which looks like it contains the names of peoples. Basically, both arrays contain the same names but data2 is a sorted version of data1

Test 1
Create a new heap file and insert records into this heap file. Each record contains two fields, the first field is from data1 and the second field is an empty string.
Then we do a scan and then perform a sort on them in ascending order of the first attribute.
Then compare the result with sorted data2. The elements should match.

Test 2
We create a new heap file but we insert records with only one attribute which is data. Next, we do a file scan and then do a sort based on attributes in descending order.
Then we compare the result with sorted data of data2 by iterating over it in reverse order. The elements should match

Test 3
Create a heap file and insert 1000 records to it which consists of 4 attributes. Then we do a file scan and sort them in ascending order based on their 3rd attribute. We get the results from the sort and compare the adjacent record to make sure that their 3rd attributes are in increasing order.
We then scan again the heap file. Then do a sort in descending order on the 4th attribute.
We get the results from the sort and compare the adjacent record to make sure that their 3rd attributes are in decreasing order.

Test 4
Here two new heap files are created and records are inserted into both heap files. The records have two attributes, where the first attribute is a string which is set to the ith element of data1 and the second attribute is empty.
We do a scan and then sort both heap files. For the first heap file, sorting is done in ascending order for the first attribute. For the second heap file, sorting is done in descending order on the first attribute.
We then compare the results of both sorts with data2. The first sort is checked data2 and the second sort results with data2 in reverse order.

Sort Merge Join Tests

The SM_JoinTest.class's main method is being executed, which results in the creation of a JoinsDriver object that extends the TestDriver class. This object is used to run join tests and is initiated by calling the runTests() method on the JoinsDriver object. The runTests() method first ends any lingering processes and then starts executing the tests. Six tests are run before the cleanup process begins.

Here we create 3 tables as below:

Sailors (sid, sname, rating, age)
Boats (bid, name, color)
Reserves(sid, bid, data)

Query1 ()
Select S.sname, R.date from Sailors as S join Reserves as R on S.sid = R.sid where R.bid = 1;

Find the sailors who have boat no. 1 reserved and the date of reservation.

Query2 ()
Empty query.

Query3 ()
Select S.sname from Sailors as S inner join Reserves as R on S.sid = R.sid;

Find the sailors who have reserved at least a boat.

Query4 ()
Select distinct S.sname from Sailors as S inner join Reserves as R on S.sid = R.sid;

Find the sailors who have reserved at least a boat and print their names exactly once.

Query5 ()
Select S.sname, S.rating, S.age from Sailors as S inner join Reserves as R on S.sid = R.sid where S.age > 40 || S.rating < 7;

Find the sailors who are more than 40 years of age or have a rating lesser than 7, and who have reserved at least a boat and print their names, ratings, and ages.

Conclusion

All the tests were successfully executed. In this phase of the project, I could understand the different components of DBMS and their functions. This project allowed me to get insights into what a relational database looked like from the inside. This phase helped me to get comfortable with minibase.

Bibliography

References:

[1]. https://research.cs.wisc.edu/coral/mini_doc/minibase.html
[2]. https://dbis.uni-konstanz.de/software/software/minibase-for-java/

Appendix

Output from my typescript file:

```
Script started on 2023-02-03 13:10:37-07:00 [TERM="xterm-256color" TTY="/dev/pts/0"
COLUMNS="156" LINES="40"]
]0;aouseph@AnjanaOuseph: /mnt/c/Users/anjana
ouseph/cse510/general.asu.edu/minjava/javaminibase/src
[01;32maouseph@AnjanaOuseph[00m:[01;34m/mnt/c/Users/anjana
ouseph/cse510/general.asu.edu/minjava/javaminibase/src[00m$ make test
cd tests; make bmtest dbtest; whoami; make hftest bttest indextest jointest sorttest sortmerge
make[1]: Entering directory '/mnt/c/Users/anjana
ouseph/cse510/general.asu.edu/minjava/javaminibase/src/tests'
/usr/bin/javac -classpath .:.. TestDriver.java BMTest.java
/usr/bin/java  -classpath .:.. tests.BMTest

Running Buffer Management tests....
Replacer: Clock


  Test 1 does a simple test of normal buffer manager operations:
  - Allocate a bunch of new pages
  - Write something on each one
  - Read that something back from each one
   (because we're buffering, this is where most of the writes happen)
  - Free the pages again
  Test 1 completed successfully.

  Test 2 exercises some illegal buffer manager operations:
  - Try to pin more pages than there are frames
*** Pinning too many pages
  --> Failed as expected

  - Try to free a doubly-pinned page
*** Freeing a pinned page
  --> Failed as expected

  - Try to unpin a page not in the buffer pool
*** Unpinning a page not in the buffer pool
  --> Failed as expected

  Test 2 completed successfully.
```

```
  Test 3 exercises some of the internals of the buffer manager
  - Allocate and dirty some new pages, one at a time, and leave some pinned
  - Read the pages
  Test 3 completed successfully.

...Buffer Management tests completely successfully.

/usr/bin/javac -classpath .:.. TestDriver.java DBTest.java
/usr/bin/java  -classpath .:.. tests.DBTest

Running Disk Space Management tests....

Replacer: Clock


  Test 1 creates a new database and does some tests of normal operations:
  - Add some file entries
  - Allocate a run of pages
  - Write something on some of them
  - Deallocate the rest of them
  Test 1 completed successfully.

  Test 2 opens the database created in test 1 and does some further tests:
  - Delete some of the file entries
  - Look up file entries that should still be there
  - Read stuff back from pages we wrote in test 1
  Test 2 completed successfully.

  Test 3 tests for some error conditions:
  - Look up a deleted file entry
**** Looking up a deleted file entry
  --> Failed as expected

  - Try to delete a deleted entry again
**** Delete a deleted file entry again
  --> Failed as expected

  - Try to delete a nonexistent file entry
**** Deleting a nonexistent file entry
  --> Failed as expected

  - Look up a nonexistent file entry
**** Looking up a nonexistent file entry
  --> Failed as expected
```

```
  - Try to add a file entry that's already there
**** Adding a duplicate file entry
  --> Failed as expected

  - Try to add a file entry whose name is too long
**** Adding a file entry with too long a name
  --> Failed as expected

  - Try to allocate a run of pages that's too long
**** Allocating a run that's too long
  --> Failed as expected

  - Try to allocate a negative run of pages
**** Allocating a negative run
  --> Failed as expected

  - Try to deallocate a negative run of pages
**** Deallocating a negative run
  --> Failed as expected

  Test 3 completed successfully.

  Test 4 tests some boundary conditions.
    (These tests are very implementation-specific.)
  - Make sure no pages are pinned
  - Allocate all pages remaining after DB overhead is accounted for
  - Attempt to allocate one more page
**** Allocating one additional page
  --> Failed as expected

  - Free some of the allocated pages
  - Allocate some of the just-freed pages
  - Free two continued run of the allocated pages
  - Allocate back number of pages equal to the just freed pages

  - Add enough file entries that the directory must surpass a page
  - Make sure that the directory has taken up an extra page: try to
    allocate more pages than should be available
**** Allocating more pages than are now available
   --> Failed as expected

  - At this point, all pages should be claimed.  Try to allocateone more.
**** Allocating one more page than there is
```

--> Failed as expected

- Free the last two pages: this tests a boundary condition in the space map.
Test 4 completed successfully.

...Disk Space Management tests completely successfully.

make[1]: Leaving directory '/mnt/c/Users/anjana
ouseph/cse510/general.asu.edu/minjava/javaminibase/src/tests'
aouseph
make[1]: Entering directory '/mnt/c/Users/anjana
ouseph/cse510/general.asu.edu/minjava/javaminibase/src/tests'
/usr/bin/javac -classpath .:.. TestDriver.java HFTest.java
/usr/bin/java -classpath .:.. tests.HFTest

Running Heap File tests....

Replacer: Clock

Test 1: Insert and scan fixed-size records

- Create a heap file

- Add 100 records to the file

- Scan the records just inserted

Test 1 completed successfully.

Test 2: Delete fixed-size records

- Open the same heap file as test 1

- Delete half the records

- Scan the remaining records

Test 2 completed successfully.

Test 3: Update fixed-size records

- Open the same heap file as tests 1 and 2

- Change the records

- Check that the updates are really there

Test 3 completed successfully.

Test 4: Test some error conditions

- Try to change the size of a record

**** Shortening a record
--> Failed as expected

**** Lengthening a record
--> Failed as expected

- Try to insert a record that's too long

**** Inserting a too-long record
--> Failed as expected

Test 4 completed successfully.

...Heap File tests completely successfully.

/usr/bin/javac -classpath .:.. TestDriver.java BTTest.java
/usr/bin/java -classpath .:.. tests.BTTest
Replacer: Clock

Running tests....

********** The file name is: AAA0 **********
---------- MENU ----------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure

[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :0
********** The file name is: AAA1 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record

[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :1
 ********** The file name is: AAA2 **********
------------------------- MENU -----------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

        ---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2
The Tree is Empty!!!
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3
The Tree is Empty!!!
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :4
Please input the page number:
3
Sorry!!! This page is neither Index nor Leaf page.
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2
The Tree is Empty!!!
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
0
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
1
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
2
------------------------- MENU -----------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

        ---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
3
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
4
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
5
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 5
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (2, [2 2])
3 (key, [pageNo, slotNo]):   (3, [3 3])
4 (key, [pageNo, slotNo]):   (4, [4 4])
5 (key, [pageNo, slotNo]):   (5, [5 5])
********** END ********

---------- All Leaf Pages Have Been Printed --------

------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
6
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 5
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 5
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (2, [2 2])
3 (key, [pageNo, slotNo]):   (3, [3 3])
4 (key, [pageNo, slotNo]):   (4, [4 4])
5 (key, [pageNo, slotNo]):   (5, [5 5])
6 (key, [pageNo, slotNo]):   (6, [6 6])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
6
Sorry!!! This page is neither Index nor Leaf page.
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

        ---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :1
********** The file name is: AAA3 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3
The Tree is Empty!!!
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :3
The Tree is Empty!!!
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
0
---------- MENU ----------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
1
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
2
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
3
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

        ---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
4
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
4
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
5
------------------------- MENU ------------------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
      and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
6
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
     and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 7
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
     and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 7
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]): (0, [0 0])

1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (2, [2 2])
3 (key, [pageNo, slotNo]): (3, [3 3])
4 (key, [pageNo, slotNo]): (4, [4 4])
5 (key, [pageNo, slotNo]): (4, [4 4])
6 (key, [pageNo, slotNo]): (5, [5 5])
7 (key, [pageNo, slotNo]): (6, [6 6])
********** END ********

---------- All Leaf Pages Have Been Printed --------

------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
4
Sorry!!! This page is neither Index nor Leaf page.
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :4
Please input the page number:
5

**********To Print an Leaf Page ********
Current Page ID: 5
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (2, [2 2])
3 (key, [pageNo, slotNo]):   (3, [3 3])
4 (key, [pageNo, slotNo]):   (4, [4 4])
5 (key, [pageNo, slotNo]):   (5, [5 5])
6 (key, [pageNo, slotNo]):   (6, [6 6])
********** END ********

------------------------- MENU ------------------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

         ---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
      and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

         ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
6
Sorry!!! This page is neither Index nor Leaf page.
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
0
Sorry!!! This page is neither Index nor Leaf page.
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:

1
Sorry!!! This page is neither Index nor Leaf page.
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
2
Sorry!!! This page is neither Index nor Leaf page.
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
3
Sorry!!! This page is neither Index nor Leaf page.
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
4
Sorry!!! This page is neither Index nor Leaf page.
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
5

**********To Print an Leaf Page ********
Current Page ID: 5
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (2, [2 2])
3 (key, [pageNo, slotNo]):   (3, [3 3])
4 (key, [pageNo, slotNo]):   (4, [4 4])
5 (key, [pageNo, slotNo]):   (5, [5 5])
6 (key, [pageNo, slotNo]):   (6, [6 6])
********** END ********

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
6
Sorry!!! This page is neither Index nor Leaf page.
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :4
Please input the page number:
7

**********To Print an Leaf Page ********
Current Page ID: 7
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (2, [2 2])
3 (key, [pageNo, slotNo]): (3, [3 3])
4 (key, [pageNo, slotNo]): (4, [4 4])
5 (key, [pageNo, slotNo]): (4, [4 4])
6 (key, [pageNo, slotNo]): (5, [5 5])
7 (key, [pageNo, slotNo]): (6, [6 6])
********** END ********

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 7
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 7
Left Link : -1
Right Link : -1

0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (2, [2 2])
3 (key, [pageNo, slotNo]): (3, [3 3])
4 (key, [pageNo, slotNo]): (4, [4 4])
5 (key, [pageNo, slotNo]): (4, [4 4])
6 (key, [pageNo, slotNo]): (5, [5 5])
7 (key, [pageNo, slotNo]): (6, [6 6])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :6
Please input the integer key to delete:
5
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 7
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (2, [2 2])
3 (key, [pageNo, slotNo]): (3, [3 3])
4 (key, [pageNo, slotNo]): (4, [4 4])
5 (key, [pageNo, slotNo]): (4, [4 4])
6 (key, [pageNo, slotNo]): (6, [6 6])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :6
Please input the integer key to delete:
4
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 7
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (2, [2 2])
3 (key, [pageNo, slotNo]): (3, [3 3])
4 (key, [pageNo, slotNo]): (4, [4 4])
5 (key, [pageNo, slotNo]): (6, [6 6])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :6
Please input the integer key to delete:
6
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order

and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 7
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (2, [2 2])
3 (key, [pageNo, slotNo]): (3, [3 3])
4 (key, [pageNo, slotNo]): (4, [4 4])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
4
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

         ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
       and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 7
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (2, [2 2])
3 (key, [pageNo, slotNo]):   (3, [3 3])
4 (key, [pageNo, slotNo]):   (4, [4 4])
5 (key, [pageNo, slotNo]):   (4, [4 4])
********** END ********

---------- All Leaf Pages Have Been Printed --------

------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :5
Please input the integer key to insert:
5
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 7
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])

2 (key, [pageNo, slotNo]): (2, [2 2])
3 (key, [pageNo, slotNo]): (3, [3 3])
4 (key, [pageNo, slotNo]): (4, [4 4])
5 (key, [pageNo, slotNo]): (4, [4 4])
6 (key, [pageNo, slotNo]): (5, [5 5])
********** END ********

---------- All Leaf Pages Have Been Printed --------

------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :6
Please input the integer key to delete:
4
------------------------- MENU ------------------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
      and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 7
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (2, [2 2])
3 (key, [pageNo, slotNo]):   (3, [3 3])
4 (key, [pageNo, slotNo]):   (4, [4 4])
5 (key, [pageNo, slotNo]):   (5, [5 5])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :7
Please input the number of keys to insert:
2
********** The file name is: AAA4 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1   9
---------- End ----------

---------- MENU ----------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
      and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 9
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
      and delete m records randomly

[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :8
Please input the number of keys to insert:
7
********** The file name is: AAA5 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 11
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 11
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (1, [1 1])
1 (key, [pageNo, slotNo]): (2, [2 2])
2 (key, [pageNo, slotNo]): (3, [3 3])
3 (key, [pageNo, slotNo]): (4, [4 4])
4 (key, [pageNo, slotNo]): (5, [5 5])
5 (key, [pageNo, slotNo]): (6, [6 6])
6 (key, [pageNo, slotNo]): (7, [7 7])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :8
Please input the number of keys to insert:
5
********** The file name is: AAA6 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record

[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 13
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (1, [1 1])
1 (key, [pageNo, slotNo]):   (2, [2 2])
2 (key, [pageNo, slotNo]):   (3, [3 3])
3 (key, [pageNo, slotNo]):   (4, [4 4])
4 (key, [pageNo, slotNo]):   (5, [5 5])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 13
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :9
Please input the number of keys to insert:
8
********** The file name is: AAA7 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 15
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 15
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (2, [2 2])
3 (key, [pageNo, slotNo]): (3, [3 3])
4 (key, [pageNo, slotNo]): (4, [4 4])

5 (key, [pageNo, slotNo]): (5, [5 5])
6 (key, [pageNo, slotNo]): (6, [6 6])
7 (key, [pageNo, slotNo]): (7, [7 7])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :9
Please input the number of keys to insert:
4
 ********** The file name is: AAA8 **********
------------------------- MENU ------------------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

          ---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
      and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

          ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
        and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 17
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (2, [2 2])
3 (key, [pageNo, slotNo]):   (3, [3 3])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :10
Please input the number of keys to insert:
8
Please input the number of keys to delete:
3
********** The file name is: AAA9 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 19
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 19
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (2, [2 2])
3 (key, [pageNo, slotNo]): (6, [6 6])
4 (key, [pageNo, slotNo]): (7, [7 7])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :11
Please input the LOWER integer key(>=0):
2
Please input the HIGHER integer key(>=0)
7
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 19
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 19
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :9
Please input the number of keys to insert:
5
********** The file name is: AAA10 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

        ---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 21
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

        ---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 21
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (2, [2 2])
3 (key, [pageNo, slotNo]):   (3, [3 3])
4 (key, [pageNo, slotNo]):   (4, [4 4])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :8
Please input the number of keys to insert:
10
********** The file name is: AAA11 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 23
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (1, [1 1])
1 (key, [pageNo, slotNo]): (2, [2 2])
2 (key, [pageNo, slotNo]): (3, [3 3])
3 (key, [pageNo, slotNo]): (4, [4 4])
4 (key, [pageNo, slotNo]): (5, [5 5])
5 (key, [pageNo, slotNo]): (6, [6 6])
6 (key, [pageNo, slotNo]): (7, [7 7])

7 (key, [pageNo, slotNo]): (8, [8 8])
8 (key, [pageNo, slotNo]): (9, [9 9])
9 (key, [pageNo, slotNo]): (10, [10 10])
********** END ********

---------- All Leaf Pages Have Been Printed --------

------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :10
Please input the number of keys to insert:
12
Please input the number of keys to delete:
2
 ********** The file name is: AAA12 **********
------------------------- MENU ------------------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

          ---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
      and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

          ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
        and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 25
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (4, [4 4])
3 (key, [pageNo, slotNo]):   (5, [5 5])
4 (key, [pageNo, slotNo]):   (6, [6 6])
5 (key, [pageNo, slotNo]):   (7, [7 7])
6 (key, [pageNo, slotNo]):   (8, [8 8])
7 (key, [pageNo, slotNo]):   (9, [9 9])
8 (key, [pageNo, slotNo]):   (10, [10 10])
9 (key, [pageNo, slotNo]):   (11, [11 11])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly

[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :11
Please input the LOWER integer key(>=0):
9
Please input the HIGHER integer key(>=0)
11
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 25
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (4, [4 4])
3 (key, [pageNo, slotNo]):   (5, [5 5])
4 (key, [pageNo, slotNo]):   (6, [6 6])
5 (key, [pageNo, slotNo]):   (7, [7 7])
6 (key, [pageNo, slotNo]):   (8, [8 8])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 25
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (4, [4 4])
3 (key, [pageNo, slotNo]): (5, [5 5])
4 (key, [pageNo, slotNo]): (6, [6 6])
5 (key, [pageNo, slotNo]): (7, [7 7])
6 (key, [pageNo, slotNo]): (8, [8 8])

********** END ********

---------- All Leaf Pages Have Been Printed --------

------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :12
Please input the LOWER integer key (null if -3):

4
Please input the HIGHER integer key (null if -2):
7
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 25
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (4, [4 4])
3 (key, [pageNo, slotNo]): (5, [5 5])
4 (key, [pageNo, slotNo]): (6, [6 6])
5 (key, [pageNo, slotNo]): (7, [7 7])
6 (key, [pageNo, slotNo]): (8, [8 8])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :13
SCAN RESULT: 4 [4 4]
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :13
SCAN RESULT: 5 [5 5]
------------------------- MENU ------------------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

         ---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
      and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

         ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
       and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :13
SCAN RESULT: 6 [6 6]
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :13
SCAN RESULT: 7 [7 7]
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :13
AT THE END OF SCAN!
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :13
AT THE END OF SCAN!
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

```
        ---String Key (for choice [15]) ---

[15]  Test5 (new file): insert n records in random order
      and delete m records randomly.

[16]  Close the file
[17]  Open which file (input an integer for the file name):
[18]  Destroy which file (input an integer for the file name):

[19]  Quit!
Hi, make your choice :`4  14
No Record to delete!
btree.ScanDeleteException
        at btree.BTFileScan.delete_current(BTFileScan.java:105)
        at tests.BTDriver.runAllTests(BTTest.java:278)
        at tests.BTDriver.runTests(BTTest.java:80)
        at tests.BTTest.main(BTTest.java:648)
btree.ScanDeleteException
        at btree.BTFileScan.delete_current(BTFileScan.java:121)
        at tests.BTDriver.runAllTests(BTTest.java:278)
        at tests.BTDriver.runTests(BTTest.java:80)
        at tests.BTTest.main(BTTest.java:648)
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    !!        Something is wrong                  !!
    !!    Is your DB full? then exit. rerun it!     !!
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
------------------------- MENU ------------------


[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

        ---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
```

[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :12
Please input the LOWER integer key (null if -3):
0
Please input the HIGHER integer key (null if -2):
8
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :13
SCAN RESULT: 0 [0 0]
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :13
SCAN RESULT: 1 [1 1]
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :13
SCAN RESULT: 4 [4 4]
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :14
------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 25
Left Link : -1
Right Link : -1
0 (key, [pageNo, slotNo]): (0, [0 0])
1 (key, [pageNo, slotNo]): (1, [1 1])
2 (key, [pageNo, slotNo]): (5, [5 5])
3 (key, [pageNo, slotNo]): (6, [6 6])
4 (key, [pageNo, slotNo]): (7, [7 7])

5 (key, [pageNo, slotNo]): (8, [8 8])
********** END ********

---------- All Leaf Pages Have Been Printed --------

------------------------- MENU ------------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :15

Please input the number of keys to insert:
8
Please input the number of keys to delete:
2
********** The file name is: AAA13 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 27
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (**1, [1 1]
1 (key, [pageNo, slotNo]):   (**2, [2 2]
2 (key, [pageNo, slotNo]):   (**4, [4 4]
3 (key, [pageNo, slotNo]):   (**5, [5 5]
4 (key, [pageNo, slotNo]):   (**6, [6 6]
5 (key, [pageNo, slotNo]):   (**7, [7 7]
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :16
********** You close the file: AAA13 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :17
12
 ********** You open the file: AAA12 **********
---------- MENU ----------

[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

        ---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
       and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :2

----------The B+ Tree Structure----------
1 25
---------- End ----------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 25
Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (0, [0 0])
1 (key, [pageNo, slotNo]):   (1, [1 1])
2 (key, [pageNo, slotNo]):   (5, [5 5])
3 (key, [pageNo, slotNo]):   (6, [6 6])
4 (key, [pageNo, slotNo]):   (7, [7 7])
5 (key, [pageNo, slotNo]):   (8, [8 8])
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :17
1
********** You open the file: AAA1 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3
The Tree is Empty!!!
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :17
13

********** You open the file: AAA13 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3

----------The B+ Tree Leaf Pages----------

**********To Print an Leaf Page ********
Current Page ID: 27

Left Link   : -1
Right Link   : -1
0 (key, [pageNo, slotNo]):   (**1, [1 1]
1 (key, [pageNo, slotNo]):   (**2, [2 2]
2 (key, [pageNo, slotNo]):   (**4, [4 4]
3 (key, [pageNo, slotNo]):   (**5, [5 5]
4 (key, [pageNo, slotNo]):   (**6, [6 6]
5 (key, [pageNo, slotNo]):   (**7, [7 7]
********** END ********

---------- All Leaf Pages Have Been Printed --------

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
     and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :18
13
********** You destroy the file: AAA13 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!

```
Hi, make your choice :17
13
 ********** You open the file: AAA13  **********
btree.ConstructPageException: pinpage failed
java.lang.NullPointerException
        at bufmgr.BufHashTbl.lookup(BufMgr.java:158)
        at bufmgr.BufMgr.pinPage(BufMgr.java:499)
        at btree.BTreeHeaderPage.<init>(BTreeHeaderPage.java:131)
        at btree.BTreeFile.<init>(BTreeFile.java:180)
        at tests.BTDriver.runAllTests(BTTest.java:300)
        at tests.BTDriver.runTests(BTTest.java:80)
        at tests.BTTest.main(BTTest.java:648)
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    !!        Something is wrong                !!
    !!    Is your DB full? then exit. rerun it!    !!
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
------------------------- MENU ------------------


[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

         ---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
[10]  Test4 (new file): insert n records in random order
     and delete m records randomly
[11]  Delete some records

[12]  Initialize a Scan
[13]  Scan the next Record
[14]  Delete the just-scanned record

         ---String Key (for choice [15]) ---

[15]  Test5 (new file): insert n records in random order
```

and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :0
********** The file name is: AAA14 **********
------------------------- MENU -----------------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :9 0
********** The file name is: AAA15 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :17
14
********** You open the file: AAA14 **********
---------- MENU ----------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :3
The Tree is Empty!!!
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
    and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

        ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
      and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :1
********** The file name is: AAA16 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

        ---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order

and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :18
16
********** You destroy the file: AAA16 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

```
         ---String Key (for choice [15]) ---

[15]  Test5 (new file): insert n records in random order
      and delete m records randomly.

[16]  Close the file
[17]  Open which file (input an integer for the file name):
[18]  Destroy which file (input an integer for the file name):

[19]  Quit!
Hi, make your choice :17
16
 ********** You open the file: AAA16  **********
btree.ConstructPageException: pinpage failed
java.lang.NullPointerException
        at bufmgr.BufHashTbl.lookup(BufMgr.java:158)
        at bufmgr.BufMgr.pinPage(BufMgr.java:499)
        at btree.BTreeHeaderPage.<init>(BTreeHeaderPage.java:131)
        at btree.BTreeFile.<init>(BTreeFile.java:180)
        at tests.BTDriver.runAllTests(BTTest.java:300)
        at tests.BTDriver.runTests(BTTest.java:80)
        at tests.BTTest.main(BTTest.java:648)
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
     !!        Something is wrong                !!
     !!    Is your DB full? then exit. rerun it!     !!
     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
------------------------- MENU -----------------


[0]   Naive delete (new file)
[1]   Full delete(Default) (new file)

[2]   Print the B+ Tree Structure
[3]   Print All Leaf Pages
[4]   Choose a Page to Print

         ---Integer Key (for choices [6]-[14]) ---

[5]   Insert a Record
[6]   Delete a Record
[7]   Test1 (new file): insert n records in order
[8]   Test2 (new file): insert n records in reverse order
[9]   Test3 (new file): insert n records in random order
```

[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :18
15
********** You destroy the file: AAA15 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :1
********** The file name is: AAA17 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :16
********** You close the file: AAA17 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :16
********** You close the file: AAA17 **********

---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :16
********** You close the file: AAA17 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record
[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :16
********** You close the file: AAA17 **********
---------- MENU ----------

[0] Naive delete (new file)
[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure
[3] Print All Leaf Pages
[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record
[6] Delete a Record

[7] Test1 (new file): insert n records in order
[8] Test2 (new file): insert n records in reverse order
[9] Test3 (new file): insert n records in random order
[10] Test4 (new file): insert n records in random order
     and delete m records randomly
[11] Delete some records

[12] Initialize a Scan
[13] Scan the next Record
[14] Delete the just-scanned record

         ---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order
       and delete m records randomly.

[16] Close the file
[17] Open which file (input an integer for the file name):
[18] Destroy which file (input an integer for the file name):

[19] Quit!
Hi, make your choice :19

... Finished .

/usr/bin/javac -classpath .:.. TestDriver.java IndexTest.java
/usr/bin/java -classpath .:.. tests.IndexTest

Running Index tests....

Replacer: Clock

---------- TEST 1 ----------
BTreeIndex created successfully.

BTreeIndex file created successfully.

Test1 -- Index Scan OK
---------- TEST 1 completed ----------

---------- TEST 2 ----------
BTreeIndex opened successfully.

Test2 -- Index Scan OK
---------- TEST 2 completed ----------

---------- TEST 3 ----------
BTreeIndex created successfully.

BTreeIndex file created successfully.

Test3 -- Index scan on int key OK

---------- TEST 3 completed ----------

...Index tests
completely successfully
.

Index tests completed successfully
/usr/bin/javac -classpath .:.. TestDriver.java JoinTest.java
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
/usr/bin/java -classpath .:.. tests.JoinTest
Replacer: Clock

Any resemblance of persons in this database to people living or dead
is purely coincidental. The contents of this database do not reflect
the views of the University, the Computer Sciences Department or the
developers...

**********Query1 strating **********
Query: Find the names of sailors who have reserved boat number 1.
    and print out the date of reservation.

```
SELECT S.sname, R.date
FROM   Sailors S, Reserves R
WHERE  S.sid = R.sid AND R.bid = 1
```

(Tests FileScan, Projection, and Sort-Merge Join)
[Mike Carey, 05/10/95]
[David Dewitt, 05/11/95]

[Jeff Naughton, 05/12/95]

Query1 completed successfully!
**********Query1 finished!!!**********

**********Query2 strating **********
Query: Find the names of sailors who have reserved a red boat
and return them in alphabetical order.

```
SELECT   S.sname
FROM     Sailors S, Boats B, Reserves R
WHERE    S.sid = R.sid AND R.bid = B.bid AND B.color = 'red'
ORDER BY S.sname
```

Plan used:
Sort (Pi(sname) (Sigma(B.color='red') |><| Pi(sname, bid) (S |><| R)))

(Tests File scan, Index scan ,Projection, index selection,
sort and simple nested-loop join.)

After Building btree index on sailors.sid.

[David Dewitt]
[Mike Carey]
[Raghu Ramakrishnan]
[Yannis Ioannidis]

Query2 completed successfully!
**********Query2 finished!!!**********

**********Query3 strating **********
Query: Find the names of sailors who have reserved a boat.

```
SELECT S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid = R.sid
```

(Tests FileScan, Projection, and SortMerge Join.)

[Mike Carey]
[Mike Carey]

[Mike Carey]
[David Dewitt]
[David Dewitt]
[Jeff Naughton]
[Miron Livny]
[Yannis Ioannidis]
[Raghu Ramakrishnan]
[Raghu Ramakrishnan]

Query3 completed successfully!
**********Query3 finished!!!**********

**********Query4 strating **********
Query: Find the names of sailors who have reserved a boat
and print each name once.

```
SELECT DISTINCT S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid = R.sid
```

(Tests FileScan, Projection, Sort-Merge Join and Duplication elimination.)

[David Dewitt]
[Jeff Naughton]
[Mike Carey]
[Miron Livny]
[Raghu Ramakrishnan]
[Yannis Ioannidis]

Query4 completed successfully!
**********Query4 finished!!!**********

**********Query5 strating **********
Query: Find the names of old sailors or sailors with a rating less
than 7, who have reserved a boat, (perhaps to increase the
amount they have to pay to make a reservation).

```
SELECT S.sname, S.rating, S.age
FROM   Sailors S, Reserves R
WHERE  S.sid = R.sid and (S.age > 40 || S.rating < 7)
```

(Tests FileScan, Multiple Selection, Projection, and Sort-Merge Join.)

[Mike Carey, 9, 40.3]
[Mike Carey, 9, 40.3]
[Mike Carey, 9, 40.3]
[David Dewitt, 10, 47.2]
[David Dewitt, 10, 47.2]
[Jeff Naughton, 5, 35.0]
[Yannis Ioannidis, 8, 40.2]

Query5 completed successfully!
**********Query5 finished!!!**********

**********Query6 strating **********
Query: Find the names of sailors with a rating greater than 7
who have reserved a red boat, and print them out in sorted order.

```
SELECT   S.sname
FROM     Sailors S, Boats B, Reserves R
WHERE    S.sid = R.sid AND S.rating > 7 AND R.bid = B.bid
         AND B.color = 'red'
ORDER BY S.name
```

Plan used:
Sort(Pi(sname) (Sigma(B.color='red') |><| Pi(sname, bid) (Sigma(S.rating > 7) |><| R)))

(Tests FileScan, Multiple Selection, Projection,sort and nested-loop join.)

After nested loop join S.sid|><|R.sid.
After nested loop join R.bid|><|B.bid AND B.color=red.
After sorting the output tuples.
[David Dewitt]
[Mike Carey]
[Raghu Ramakrishnan]
[Yannis Ioannidis]

Query6 completed successfully!
**********Query6 finished!!!**********

```
Finished joins testing
join tests completed successfully
/usr/bin/javac -classpath .:.. TestDriver.java SortTest.java
/usr/bin/java  -classpath .:.. tests.SortTest

Running Sort tests....

Replacer: Clock

---------------------- TEST 1 -------------------------
Test1 -- Sorting OK
------------------- TEST 1 completed --------------------

---------------------- TEST 2 -------------------------
Test2 -- Sorting OK
------------------- TEST 2 completed --------------------

---------------------- TEST 3 -------------------------
 -- Sorting in ascending order on the int field --
Test3 -- Sorting of int field OK

 -- Sorting in descending order on the float field --
Test3 -- Sorting of float field OK

------------------- TEST 3 completed --------------------

---------------------- TEST 4 -------------------------
Test4 -- Sorting OK
------------------- TEST 4 completed --------------------


...Sort tests
completely successfully
.


Sorting tests completed successfully
/usr/bin/javac -classpath .:.. SM_JoinTest.java TestDriver.java
Note: SM_JoinTest.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
/usr/bin/java  -classpath .:.. tests.SM_JoinTest
Replacer: Clock
```

Any resemblance of persons in this database to people living or dead
is purely coincidental. The contents of this database do not reflect
the views of the University, the Computer Sciences Department or the
developers...

**********Query1 strating **********
Query: Find the names of sailors who have reserved boat number 1.
      and print out the date of reservation.

```
SELECT S.sname, R.date
FROM   Sailors S, Reserves R
WHERE  S.sid = R.sid AND R.bid = 1
```

(Tests FileScan, Projection, and Sort-Merge Join)
[Mike Carey, 05/10/95]
[David Dewitt, 05/11/95]
[Jeff Naughton, 05/12/95]

Query1 completed successfully!
**********Query1 finished!!!**********

**********Query3 strating **********
Query: Find the names of sailors who have reserved a boat.

```
SELECT S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid = R.sid
```

(Tests FileScan, Projection, and SortMerge Join.)

[Mike Carey]
[Mike Carey]
[Mike Carey]
[David Dewitt]
[David Dewitt]
[Jeff Naughton]
[Miron Livny]
[Yannis Ioannidis]
[Raghu Ramakrishnan]
[Raghu Ramakrishnan]

Query3 completed successfully!
**********Query3 finished!!!**********

**********Query4 strating **********
Query: Find the names of sailors who have reserved a boat
and print each name once.

```
SELECT DISTINCT S.sname
FROM   Sailors S, Reserves R
WHERE  S.sid = R.sid
```

(Tests FileScan, Projection, Sort-Merge Join and Duplication elimination.)

[David Dewitt]
[Jeff Naughton]
[Mike Carey]
[Miron Livny]
[Raghu Ramakrishnan]
[Yannis Ioannidis]

Query4 completed successfully!
**********Query4 finished!!!**********

**********Query5 strating **********
Query: Find the names of old sailors or sailors with a rating less
than 7, who have reserved a boat, (perhaps to increase the
amount they have to pay to make a reservation).

```
SELECT S.sname, S.rating, S.age
FROM   Sailors S, Reserves R
WHERE  S.sid = R.sid and (S.age > 40 || S.rating < 7)
```

(Tests FileScan, Multiple Selection, Projection, and Sort-Merge Join.)

[Mike Carey, 9, 40.3]
[Mike Carey, 9, 40.3]
[Mike Carey, 9, 40.3]
[David Dewitt, 10, 47.2]
[David Dewitt, 10, 47.2]

[Jeff Naughton, 5, 35.0]
[Yannis Ioannidis, 8, 40.2]

Query5 completed successfully!
**********Query5 finished!!!**********

Finished joins testing
join tests completed successfully
make[1]: Leaving directory '/mnt/c/Users/anjana
ouseph/cse510/general.asu.edu/minjava/javaminibase/src/tests'
]0;aouseph@AnjanaOuseph: /mnt/c/Users/anjana
ouseph/cse510/general.asu.edu/minjava/javaminibase/src
[01;32maouseph@AnjanaOuseph[00m:[01;34m/mnt/c/Users/anjana
ouseph/cse510/general.asu.edu/minjava/javaminibase/src[00m$ exit

Script done on 2023-02-03 20:11:03-07:00 [COMMAND_EXIT_CODE="0"]