# B+ TREE IMPLEMENTATION

Using MINIBASE

# Table of Contents

# Abstract

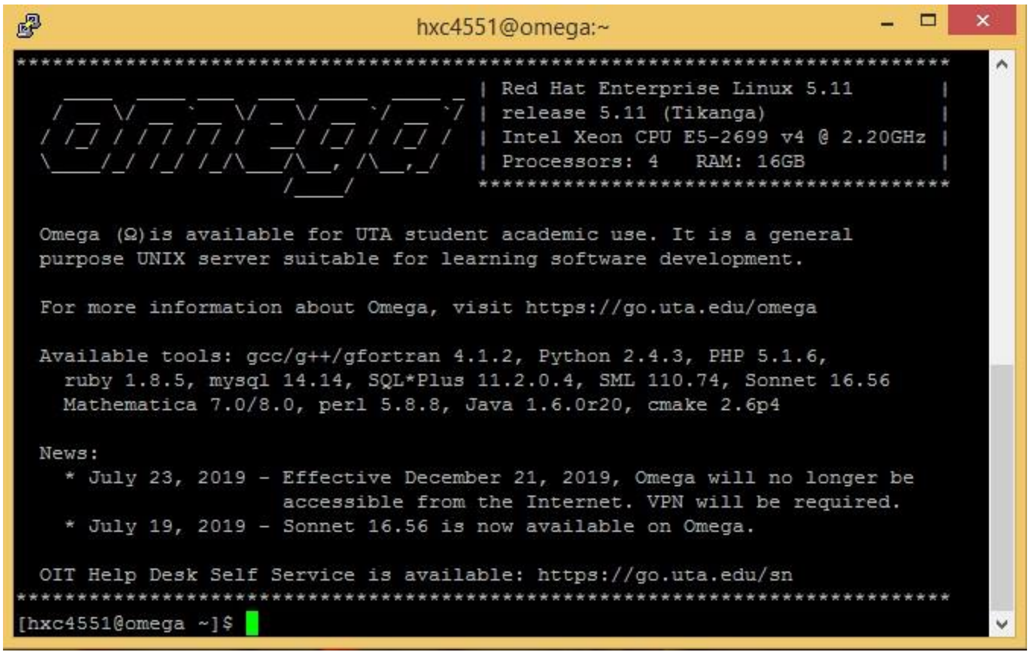
In this project, we have implemented the insert and delete functions for B+ tree. The insert function implemented is capable of dealing with overflows by splitting. The delete function on the other hand, is naïve delete and does not deal with merging i.e. it simply deletes the record from the tree. Thus, we have implemented the methods insert(), _insert() and NaiveDelete().

# Overall Status

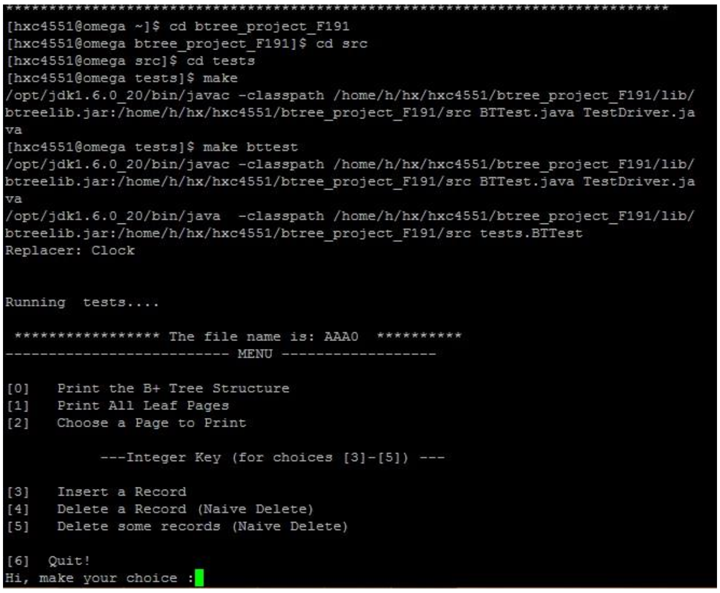
## Compiling the files

We used the omega server for remotely accessing the files assigned with our MavId account. And we used it for the purpose of implementing this project as mentioned by the attachment provided by Sir and TA. Omega server is accessible only with on-campus Wi-fi. So during off-campus times, it made us strictly code with eclipse environment which was using jdk 1.8 compiler version, and which was not the version omega was using. Due to this, we encountered many problems like version clash (like 50 and 52 class error) of files while compiling the files in omega. But with eclipse environment, it was running fine.

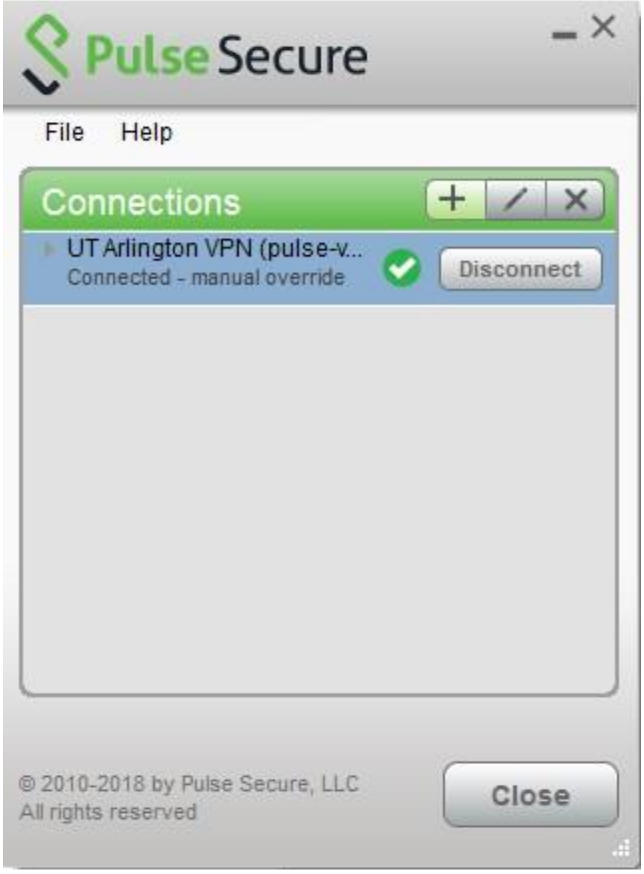
To overcome this problem, we made use of VPN, and NetIDplus as mentioned on the omega.uta.edu website. And we installed Duo Mobile, Pulse Secure which made us work during our off-campus hours. With this we were able to test it easily anytime with no issues whatsoever. Putty was used as a command line interface to work with remote host.



```
****************************************************************************
   _____ ____ ____ ____ ____ ____ _  | Red Hat Enterprise Linux 5.11        |
  / __ \/ __ `__ \/ _ \/ __ `/ __ `/ | release 5.11 (Tikanga)               |
 / /_/ / / / / / /  __/ /_/ / /_/ /  | Intel Xeon CPU E5-2699 v4 @ 2.20GHz |
 \____/_/ /_/ /_/\___/\__, /\__,_/   | Processors: 4   RAM: 16GB           |
                     /____/          ****************************************

  Omega (Ω)is available for UTA student academic use. It is a general
  purpose UNIX server suitable for learning software development.

  For more information about Omega, visit https://go.uta.edu/omega

  Available tools: gcc/g++/gfortran 4.1.2, Python 2.4.3, PHP 5.1.6,
    ruby 1.8.5, mysql 14.14, SQL*Plus 11.2.0.4, SML 110.74, Sonnet 16.56
    Mathematica 7.0/8.0, perl 5.8.8, Java 1.6.0r20, cmake 2.6p4

  News:
    * July 23, 2019 - Effective December 21, 2019, Omega will no longer be
                      accessible from the Internet. VPN will be required.
    * July 19, 2019 - Sonnet 16.56 is now available on Omega.

  OIT Help Desk Self Service is available: https://go.uta.edu/sn
****************************************************************************
[hxc4551@omega ~]$
```

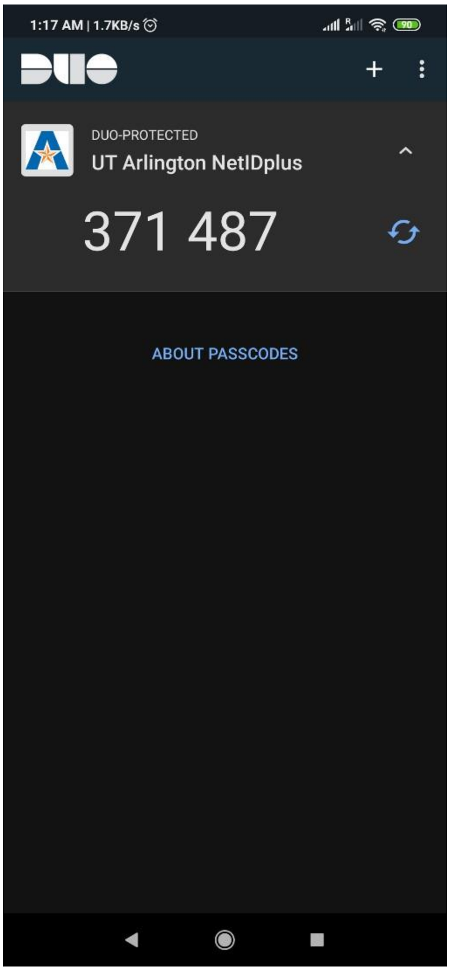


```
*****************************************************************************
[hxc4551@omega ~]$ cd btree_project_F191
[hxc4551@omega btree_project_F191]$ cd src
[hxc4551@omega src]$ cd tests
[hxc4551@omega tests]$ make
/opt/jdk1.6.0_20/bin/javac -classpath /home/h/hx/hxc4551/btree_project_F191/lib/
btreelib.jar:/home/h/hx/hxc4551/btree_project_F191/src BTTest.java TestDriver.ja
va
[hxc4551@omega tests]$ make bttest
/opt/jdk1.6.0_20/bin/javac -classpath /home/h/hx/hxc4551/btree_project_F191/lib/
btreelib.jar:/home/h/hx/hxc4551/btree_project_F191/src BTTest.java TestDriver.ja
va
/opt/jdk1.6.0_20/bin/java  -classpath /home/h/hx/hxc4551/btree_project_F191/lib/
btreelib.jar:/home/h/hx/hxc4551/btree_project_F191/src tests.BTTest
Replacer: Clock


Running  tests....

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

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

          ---Integer Key (for choices [3]-[5]) ---

[3]   Insert a Record
[4]   Delete a Record (Naive Delete)
[5]   Delete some records (Naive Delete)

[6]  Quit!
Hi, make your choice :
```



UT Arlington VPN (pulse-v...
Connected - manual override

© 2010-2018 by Pulse Secure, LLC
All rights reserved



DUO-PROTECTED
UT Arlington NetIDplus
371 487

<u>Pulse Secure : Windows client</u>

This helps to connect omega.uta.edu (host) through a secure lin:e without any issues whatsoever.

<u>DUO Mobile Client</u>

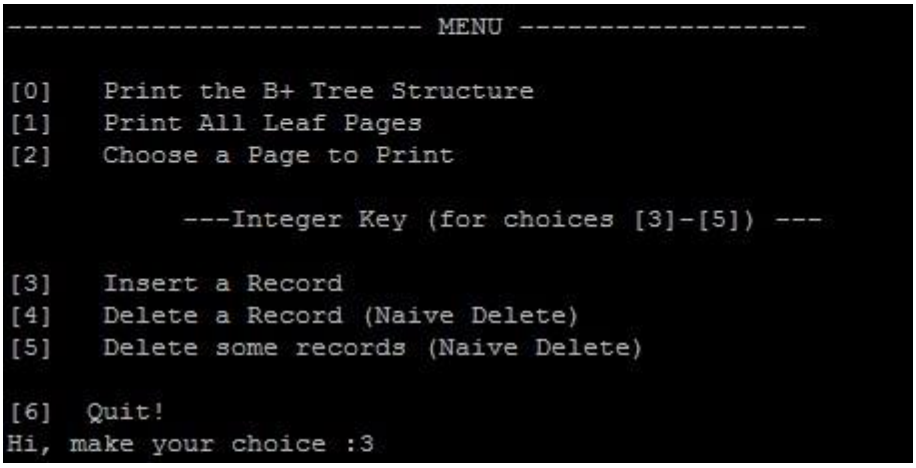
Duo Mobile app helps you generate realtime NetIDplus code to allow Pulse Secure to connect omega.uta.edu easily with authentication.

# Insert Method

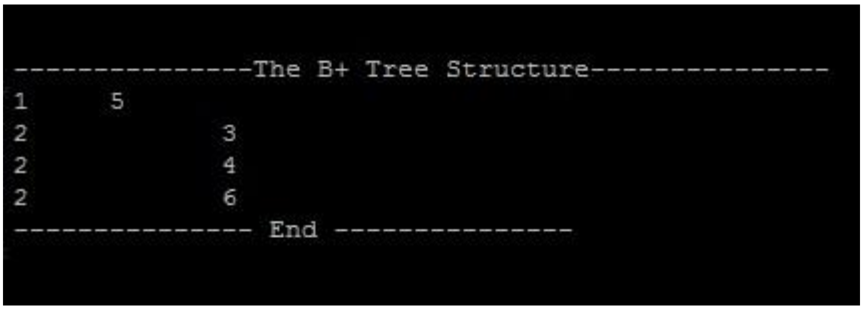
The insert() method takes the parameters <key,RID>. The B+ tree will initially be empty and the header will point to INVALID_PAGE. When the first record is inserted, a new leaf page is created and the header is made to point to this new page.

All other insertion operations will happen via the _insert() method that takes parameters <key, RID, currentPageID>. This function checks if the entry is a leaf or index page and in case of leaf page, it checks if there's space. If the there is no space then it splits the data into a new leaf page and creates an index that maps to the newly created leaf. It is found that in this project, the page splits when the 63rd record is entered. The existing page takes the first 31 records and the newly created page takes the remaining 31 records.

Initially, we entered 100 records and the following structure of tree was obtained

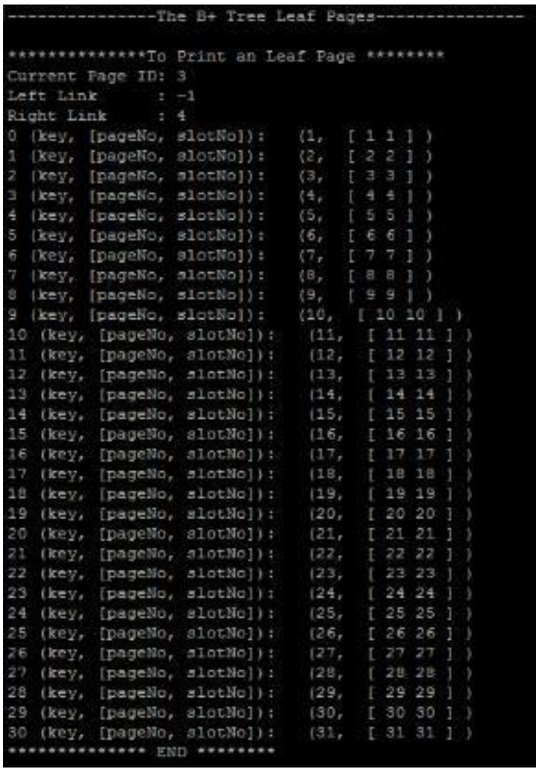


```
-------------------------- MENU ------------------

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

          ---Integer Key (for choices [3]-[5]) ---

[3]    Insert a Record
[4]    Delete a Record (Naive Delete)
[5]    Delete some records (Naive Delete)

[6]   Quit!
Hi, make your choice :3
```

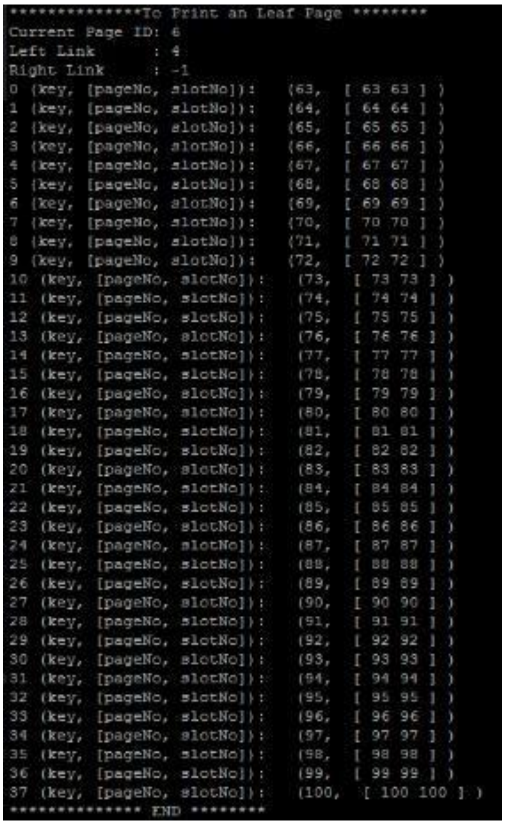


```
---------------The B+ Tree Structure---------------
1     5
2            3
2            4
2            6
--------------- End ---------------
```

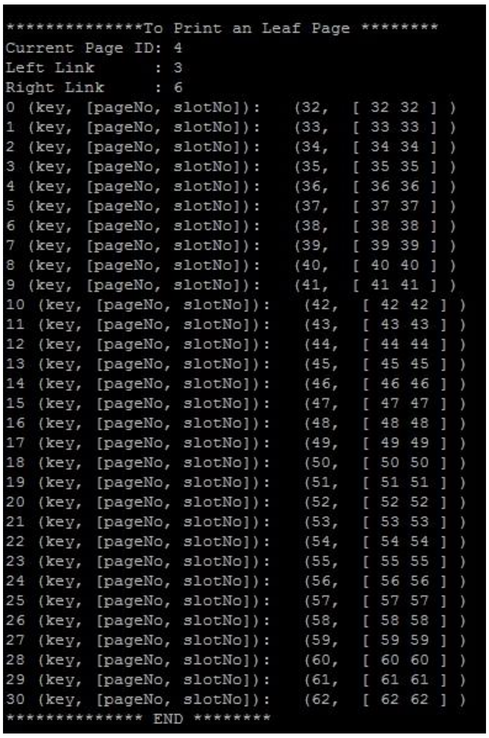
The first, second & last page of the tree are as follows :



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

**************To Print an Leaf Page ********
Current Page ID: 3
Left Link      : -1
Right Link     : 4
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 ] )
10 (key, [pageNo, slotNo]):   (11,  [ 11 11 ] )
11 (key, [pageNo, slotNo]):   (12,  [ 12 12 ] )
12 (key, [pageNo, slotNo]):   (13,  [ 13 13 ] )
13 (key, [pageNo, slotNo]):   (14,  [ 14 14 ] )
14 (key, [pageNo, slotNo]):   (15,  [ 15 15 ] )
15 (key, [pageNo, slotNo]):   (16,  [ 16 16 ] )
16 (key, [pageNo, slotNo]):   (17,  [ 17 17 ] )
17 (key, [pageNo, slotNo]):   (18,  [ 18 18 ] )
18 (key, [pageNo, slotNo]):   (19,  [ 19 19 ] )
19 (key, [pageNo, slotNo]):   (20,  [ 20 20 ] )
20 (key, [pageNo, slotNo]):   (21,  [ 21 21 ] )
21 (key, [pageNo, slotNo]):   (22,  [ 22 22 ] )
22 (key, [pageNo, slotNo]):   (23,  [ 23 23 ] )
23 (key, [pageNo, slotNo]):   (24,  [ 24 24 ] )
24 (key, [pageNo, slotNo]):   (25,  [ 25 25 ] )
25 (key, [pageNo, slotNo]):   (26,  [ 26 26 ] )
26 (key, [pageNo, slotNo]):   (27,  [ 27 27 ] )
27 (key, [pageNo, slotNo]):   (28,  [ 28 28 ] )
28 (key, [pageNo, slotNo]):   (29,  [ 29 29 ] )
29 (key, [pageNo, slotNo]):   (30,  [ 30 30 ] )
30 (key, [pageNo, slotNo]):   (31,  [ 31 31 ] )
************** END ********
```



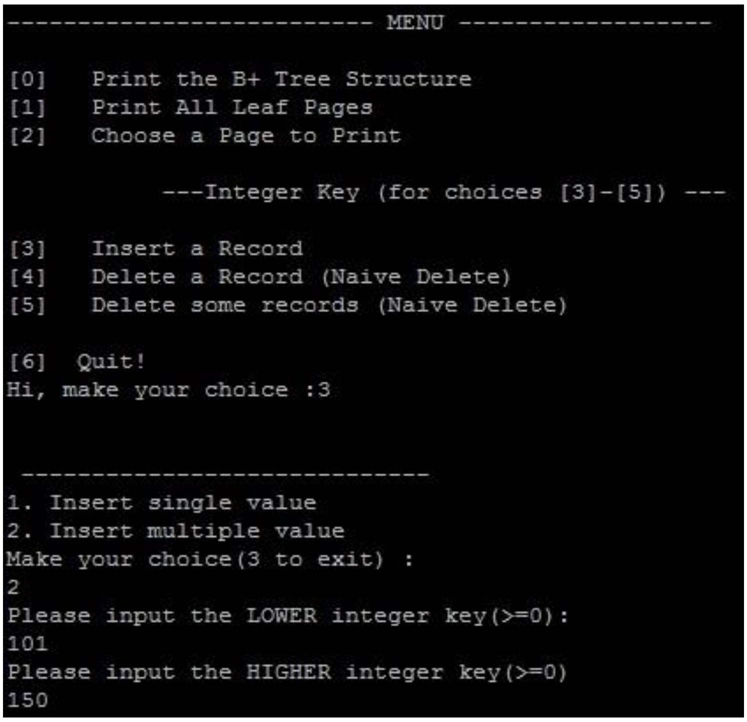
```
**************To Print an Leaf Page ********
Current Page ID: 6
Left Link      : 4
Right Link     : -1
0 (key, [pageNo, slotNo]):   (63,   [ 63 63 ] )
1 (key, [pageNo, slotNo]):   (64,   [ 64 64 ] )
2 (key, [pageNo, slotNo]):   (65,   [ 65 65 ] )
3 (key, [pageNo, slotNo]):   (66,   [ 66 66 ] )
4 (key, [pageNo, slotNo]):   (67,   [ 67 67 ] )
5 (key, [pageNo, slotNo]):   (68,   [ 68 68 ] )
6 (key, [pageNo, slotNo]):   (69,   [ 69 69 ] )
7 (key, [pageNo, slotNo]):   (70,   [ 70 70 ] )
8 (key, [pageNo, slotNo]):   (71,   [ 71 71 ] )
9 (key, [pageNo, slotNo]):   (72,   [ 72 72 ] )
10 (key, [pageNo, slotNo]):   (73,   [ 73 73 ] )
11 (key, [pageNo, slotNo]):   (74,   [ 74 74 ] )
12 (key, [pageNo, slotNo]):   (75,   [ 75 75 ] )
13 (key, [pageNo, slotNo]):   (76,   [ 76 76 ] )
14 (key, [pageNo, slotNo]):   (77,   [ 77 77 ] )
15 (key, [pageNo, slotNo]):   (78,   [ 78 78 ] )
16 (key, [pageNo, slotNo]):   (79,   [ 79 79 ] )
17 (key, [pageNo, slotNo]):   (80,   [ 80 80 ] )
18 (key, [pageNo, slotNo]):   (81,   [ 81 81 ] )
19 (key, [pageNo, slotNo]):   (82,   [ 82 82 ] )
20 (key, [pageNo, slotNo]):   (83,   [ 83 83 ] )
21 (key, [pageNo, slotNo]):   (84,   [ 84 84 ] )
22 (key, [pageNo, slotNo]):   (85,   [ 85 85 ] )
23 (key, [pageNo, slotNo]):   (86,   [ 86 86 ] )
24 (key, [pageNo, slotNo]):   (87,   [ 87 87 ] )
25 (key, [pageNo, slotNo]):   (88,   [ 88 88 ] )
26 (key, [pageNo, slotNo]):   (89,   [ 89 89 ] )
27 (key, [pageNo, slotNo]):   (90,   [ 90 90 ] )
28 (key, [pageNo, slotNo]):   (91,   [ 91 91 ] )
29 (key, [pageNo, slotNo]):   (92,   [ 92 92 ] )
30 (key, [pageNo, slotNo]):   (93,   [ 93 93 ] )
31 (key, [pageNo, slotNo]):   (94,   [ 94 94 ] )
32 (key, [pageNo, slotNo]):   (95,   [ 95 95 ] )
33 (key, [pageNo, slotNo]):   (96,   [ 96 96 ] )
34 (key, [pageNo, slotNo]):   (97,   [ 97 97 ] )
35 (key, [pageNo, slotNo]):   (98,   [ 98 98 ] )
36 (key, [pageNo, slotNo]):   (99,   [ 99 99 ] )
37 (key, [pageNo, slotNo]):   (100,  [ 100 100 ] )
************** END ********
```



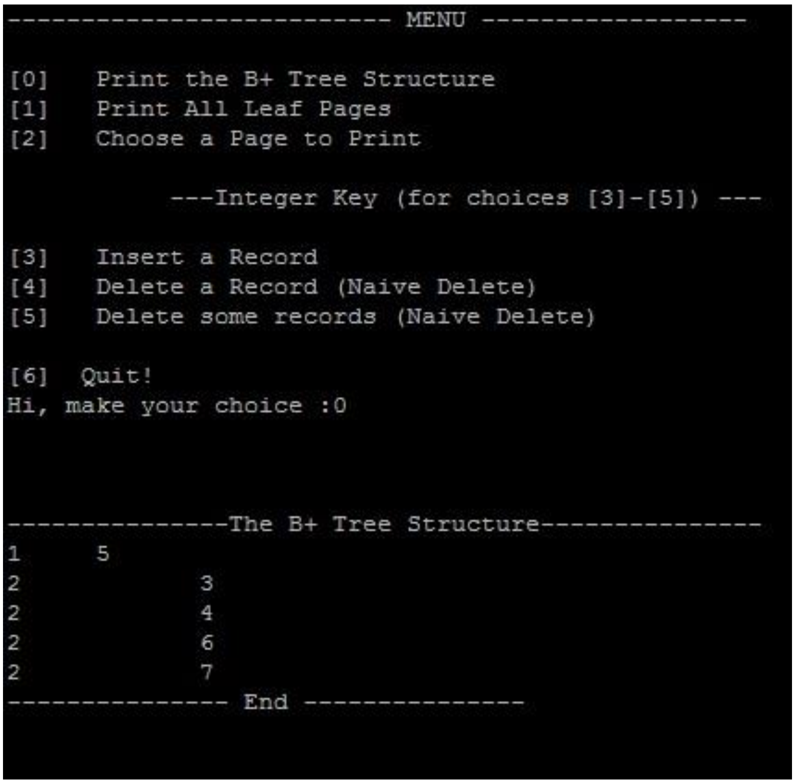
```
**************To Print an Leaf Page ********
Current Page ID: 4
Left Link      : 3
Right Link     : 6
0 (key, [pageNo, slotNo]):   (32,   [ 32 32 ] )
1 (key, [pageNo, slotNo]):   (33,   [ 33 33 ] )
2 (key, [pageNo, slotNo]):   (34,   [ 34 34 ] )
3 (key, [pageNo, slotNo]):   (35,   [ 35 35 ] )
4 (key, [pageNo, slotNo]):   (36,   [ 36 36 ] )
5 (key, [pageNo, slotNo]):   (37,   [ 37 37 ] )
6 (key, [pageNo, slotNo]):   (38,   [ 38 38 ] )
7 (key, [pageNo, slotNo]):   (39,   [ 39 39 ] )
8 (key, [pageNo, slotNo]):   (40,   [ 40 40 ] )
9 (key, [pageNo, slotNo]):   (41,   [ 41 41 ] )
10 (key, [pageNo, slotNo]):   (42,   [ 42 42 ] )
11 (key, [pageNo, slotNo]):   (43,   [ 43 43 ] )
12 (key, [pageNo, slotNo]):   (44,   [ 44 44 ] )
13 (key, [pageNo, slotNo]):   (45,   [ 45 45 ] )
14 (key, [pageNo, slotNo]):   (46,   [ 46 46 ] )
15 (key, [pageNo, slotNo]):   (47,   [ 47 47 ] )
16 (key, [pageNo, slotNo]):   (48,   [ 48 48 ] )
17 (key, [pageNo, slotNo]):   (49,   [ 49 49 ] )
18 (key, [pageNo, slotNo]):   (50,   [ 50 50 ] )
19 (key, [pageNo, slotNo]):   (51,   [ 51 51 ] )
20 (key, [pageNo, slotNo]):   (52,   [ 52 52 ] )
21 (key, [pageNo, slotNo]):   (53,   [ 53 53 ] )
22 (key, [pageNo, slotNo]):   (54,   [ 54 54 ] )
23 (key, [pageNo, slotNo]):   (55,   [ 55 55 ] )
24 (key, [pageNo, slotNo]):   (56,   [ 56 56 ] )
25 (key, [pageNo, slotNo]):   (57,   [ 57 57 ] )
26 (key, [pageNo, slotNo]):   (58,   [ 58 58 ] )
27 (key, [pageNo, slotNo]):   (59,   [ 59 59 ] )
28 (key, [pageNo, slotNo]):   (60,   [ 60 60 ] )
29 (key, [pageNo, slotNo]):   (61,   [ 61 61 ] )
30 (key, [pageNo, slotNo]):   (62,   [ 62 62 ] )
************** END ********
```

The first, second & last page are displayed in order demonstrate the left link and the right link.

Furthermore, we insert 50 more entries to see the change in the b+tree structure as follows:



```
-------------------------- MENU ------------------

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

        ---Integer Key (for choices [3]-[5]) ---

[3]   Insert a Record
[4]   Delete a Record (Naive Delete)
[5]   Delete some records (Naive Delete)

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


 -----------------------------
1. Insert single value
2. Insert multiple value
Make your choice(3 to exit) :
2
Please input the LOWER integer key(>=0):
101
Please input the HIGHER integer key(>=0)
150
```



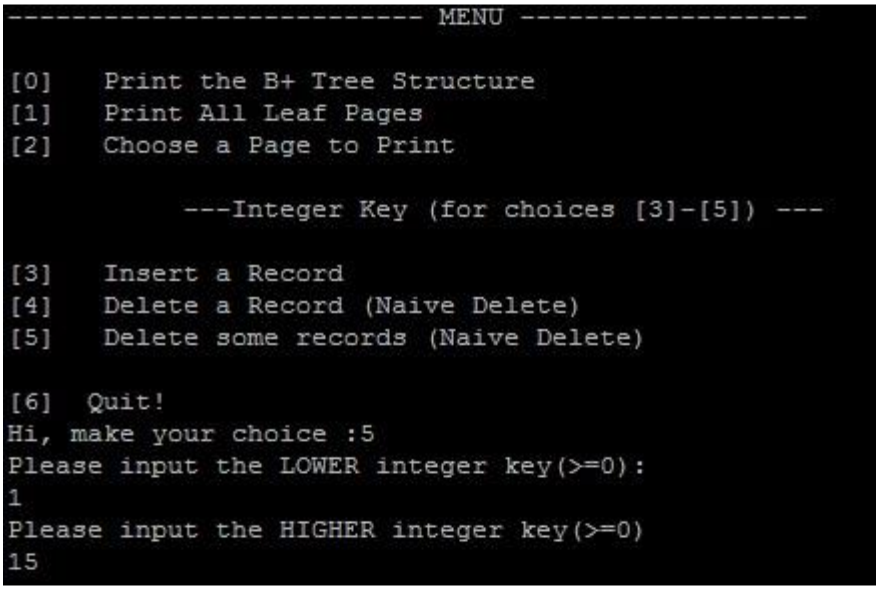
```
-------------------------- MENU ------------------

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

        ---Integer Key (for choices [3]-[5]) ---

[3]   Insert a Record
[4]   Delete a Record (Naive Delete)
[5]   Delete some records (Naive Delete)

[6]  Quit!
Hi, make your choice :0



---------------The B+ Tree Structure---------------
1     5
2             3
2             4
2             6
2             7
--------------- End ---------------
```

Here, we can clearly see that after adding 50 new entries into the structure, the tree structure changes with PageId 7 coming into picture.

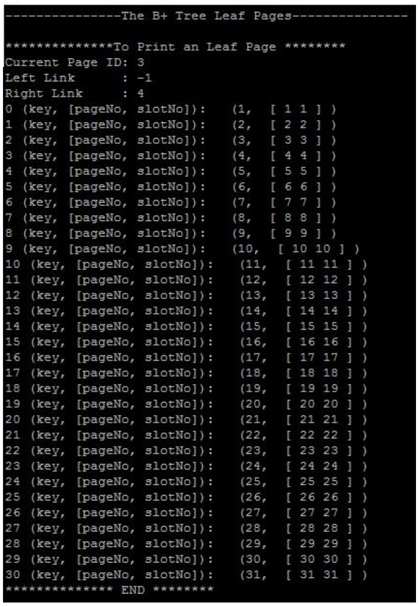
## Delete Method

In this project, with the NaiveDelete() method, we deleted the record directly from the tree without balancing the tree i.e. it won't take into account for merging or redistribution of records after deletion. The NaiveDelete() function will take the parameters <key, RID>. We will use the key to traverse the tree and compare the key value. Once we find the match, we delete that record.

On deleting the first 15 records from the existing tree, we get the following output



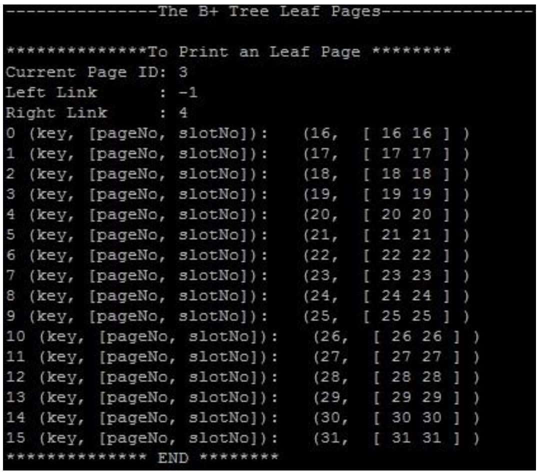
```
-------------------------- MENU -------------------

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

           ---Integer Key (for choices [3]-[5]) ---

[3]   Insert a Record
[4]   Delete a Record (Naive Delete)
[5]   Delete some records (Naive Delete)

[6]  Quit!
Hi, make your choice :5
Please input the LOWER integer key(>=0):
1
Please input the HIGHER integer key(>=0)
15
```



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

**************To Print an Leaf Page ********
Current Page ID: 3
Left Link      : -1
Right Link     : 4
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 ] )
10 (key, [pageNo, slotNo]):   (11,  [ 11 11 ] )
11 (key, [pageNo, slotNo]):   (12,  [ 12 12 ] )
12 (key, [pageNo, slotNo]):   (13,  [ 13 13 ] )
13 (key, [pageNo, slotNo]):   (14,  [ 14 14 ] )
14 (key, [pageNo, slotNo]):   (15,  [ 15 15 ] )
15 (key, [pageNo, slotNo]):   (16,  [ 16 16 ] )
16 (key, [pageNo, slotNo]):   (17,  [ 17 17 ] )
17 (key, [pageNo, slotNo]):   (18,  [ 18 18 ] )
18 (key, [pageNo, slotNo]):   (19,  [ 19 19 ] )
19 (key, [pageNo, slotNo]):   (20,  [ 20 20 ] )
20 (key, [pageNo, slotNo]):   (21,  [ 21 21 ] )
21 (key, [pageNo, slotNo]):   (22,  [ 22 22 ] )
22 (key, [pageNo, slotNo]):   (23,  [ 23 23 ] )
23 (key, [pageNo, slotNo]):   (24,  [ 24 24 ] )
24 (key, [pageNo, slotNo]):   (25,  [ 25 25 ] )
25 (key, [pageNo, slotNo]):   (26,  [ 26 26 ] )
26 (key, [pageNo, slotNo]):   (27,  [ 27 27 ] )
27 (key, [pageNo, slotNo]):   (28,  [ 28 28 ] )
28 (key, [pageNo, slotNo]):   (29,  [ 29 29 ] )
29 (key, [pageNo, slotNo]):   (30,  [ 30 30 ] )
30 (key, [pageNo, slotNo]):   (31,  [ 31 31 ] )
************** END ********
```

BEFORE

AFTER



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

**************To Print an Leaf Page ********
Current Page ID: 3
Left Link      : -1
Right Link     : 4
0 (key, [pageNo, slotNo]):   (16,  [ 16 16 ] )
1 (key, [pageNo, slotNo]):   (17,  [ 17 17 ] )
2 (key, [pageNo, slotNo]):   (18,  [ 18 18 ] )
3 (key, [pageNo, slotNo]):   (19,  [ 19 19 ] )
4 (key, [pageNo, slotNo]):   (20,  [ 20 20 ] )
5 (key, [pageNo, slotNo]):   (21,  [ 21 21 ] )
6 (key, [pageNo, slotNo]):   (22,  [ 22 22 ] )
7 (key, [pageNo, slotNo]):   (23,  [ 23 23 ] )
8 (key, [pageNo, slotNo]):   (24,  [ 24 24 ] )
9 (key, [pageNo, slotNo]):   (25,  [ 25 25 ] )
10 (key, [pageNo, slotNo]):   (26,  [ 26 26 ] )
11 (key, [pageNo, slotNo]):   (27,  [ 27 27 ] )
12 (key, [pageNo, slotNo]):   (28,  [ 28 28 ] )
13 (key, [pageNo, slotNo]):   (29,  [ 29 29 ] )
14 (key, [pageNo, slotNo]):   (30,  [ 30 30 ] )
15 (key, [pageNo, slotNo]):   (31,  [ 31 31 ] )
************** END ********
```

# File Description

No new files were created in this projects. All files used were provided in the assignment attachment. In the assignment, minibase library was given which was used for referencing classes and objects to make the insert(), _insert() & NaiveDelete().

# Division of Labor

The initial days were given to understand the various functions used in MiniBase by reading the documentation. Later, we together learnt the concepts of B+ tree and formulated the logic that we would use to implement the insert and delete functions. The code for insert() function was done by Ankita and the code for NaïveDelete() function was done by Harsh while the _insert() function was done by both of us.

# Logical Errors

1) With minibase library, it was easy to decode the java methods and classes mentioned. But to set-up the methods was a little too challenging. And, everything was predefined. For example, if you wanted to return something like false/true, normally you can just use return 0/1 to do the job but we later got to know that it was a pre-set to Boolean value and it should either be true or false.

2) Also, while splitting, simple mistakes like not setting the nextPage and previousPage to null or different pages led to many problems.

3) While making the B+ tree structure, we encountered our mistake such as assuming the first node as index node. We forgot that the first node taken has to be a leaf node and when it is full we make an index node and then add the leaf nodes respectively with the process of splitting.

4) In the beginning, we thought that only the rightmost node was splitting. And this made us always point the nextPage to null for the newsplitleaf. This gave us trouble during insertion of duplicate entries.

5) InvalidSlotNumberException – This was one of the first problem we encountered while checking the header exists or not. We later on found that we were using it wrong with _insert().

6) Also, with the NaiveDelete() method, we first thought that merging and re-distribution had to be implemented. This gave us a lot of problems while dividing tasks between us group members. And this messed up our timeline chart. Later on, after talking with TA at the time of 1st milestone during the feedback, we got to know that only deletion of records had to be implemented.