`CSE 5331 : B+ Tree Final Submission`

```
Group No. 17
Team Members:
1. Rahul Thapar
2. Harrish Murugesan
```

Overall Status

> Completed: `insert()`, `_insert()` and `naiveDelete()`.

`Approach for milestone 1:`

1. Went through the Minibase library to understand the functions and their usage.
2. Followed `Incremental Development` technique through out.
3. Things were divided among the 2 group members to develop and test as the process progressed.
4. BTTest.java file was used as the test case file.

`Approach for final submission:`

1. Deleting a particular record (or) a series of records were accomplished.
2. Since we are handling a Naïve form of deletion, redistribution of nodes were not touched upon.
3. Naive Delete was developed and tested by dry run of the code and then with the test cases.
4. BTTest.java file was used to test the correctness of the code.

`Things Achieved:`

1. Implemented insert() and _insert().
2. Able to insert 100 values.
3. Able to split the fully filled node.

```
Incase for INDEX node : moveUP was done
Incase for LEAF node : copyUP was done
```

4. Complete implementation of the NaiveDelete().
5. Able to locate and delete a particular record from the B+ Tree.
6. Able to successfully delete a range of records.

File Description

No new file was created.

Division of Labor

1. Rahul Thapar : _insert(LEAF), naiveDelete(), testing (dry run)
2. Harrish Murugesan: insert(), _insert(INDEX), testing (dry run + test cases from BTTest.java)
3. `Agile` process framework was followed by conducting daily scrum meetings to keep track of what progress was made and what problems were faced.

Errors Encountered

```
 1. Multiple roots being created (SOLVED)
------------------------------------------

- Multiple roots were created while implementing the insert() function.

- This was caused because the initialisation of the root page was done before
the Invalid page check. Hence, it created a new root every time and then checked
if the header has to be created at the first place or not.

- This was resolved by moving the initialising part inside the Invalid page
if condition so that it would be created only when the tree did not exist
that is root would be created only once.
```

```
 2. BufferManagement Issue (SOLVED)
------------------------------------------

- While inserting the value in the index node we ran into a buffer
management issue which said that the page was not pinned and we were
trying to unpin the page.

- This was caused because we created the new index page and passed the wrong id.
Thus, while unpinning the page the function couldn't resolve the correct id
which we provided. The actual page was never pinned, hence causing the issue.

- This was resolved by initialising the new index page with correct pageid.
```

```

3. Deleting a Range of records (SOLVED):
------------------------------------------

- When proceeding to delete a range of records after a particular record was
deleted was causing an error.

- This was caused because while deleting, unpining the page was done. When a
range of values were deleted and the pointer pointed to the previously deleted
value, the reference sent to unpinPage() did not exist (since it was already deleted
before), hence causing the issue.

- This was resolved by getting the correct reference to the next page after a page
has been deleted to avoid creating a wrong reference for the next iteration.

```

`Conclusions drawn from the results:`

`Milestone 1`

1. `62` records get entered before the first split happens.
2. Order of the tree formed is `2`.

`Final Submission`

1. Index split occurs when the page ID reaches `89`.
2. Order of the tree formed is `2`.
