Lab 2 Write up

U34502501 Haoran Wei

1. Design decisions:

   For the bufferpool implementation, I have completed the insertTuple and deleteTuple and pageEviction. Especially for the pageEviction, I have created a LRU (Least Recently Used) data structure called recenltyUsed which will be updated once a page is called. So in this way, the biggest number in the recenlyUsed will be moved when the bufferPages are full.

   Insertion methods in B+ tree：

   For those method, I have to read a lot of the javadoc and the code structure. After that, the implementation becomes easier. The only trick is that the statement hasn't listed that bufferpool is a must to implement so that I spent a lot of time finding nothing. The thing more than that is the deletions and insertion sequence, because insertion will change the recordId, if insertion comes first, the deletions will find nothing.

   Idea in soving bonus:

   Deletion is more complex than the insertion, but both have roughly the same procedure. The difficult one is how to move the internal page up and down one by one. I spent a lot of time to make sure the delete whether left or right child and how to maintain a correct B plus tree relationship.

2. Changes to API: None

3. missing or incomplete part : None.
4. Time spent on the lab : 12 hours