Descriptions of Lab 2

This is a required description file for lab 2 by **Zhanghao Wu** (516030910593)

Design decisions

1. **Eviction Policy**: In this lab, I use LRU as eviction policy. Using a Hashmap *LRUCount* (a saturating counter for each page), I keep track of the usage of each page. Whenever a page is read or write in the buffer, the counter will be set to limitation of the buffer for this page, and decreased by 1 (will not be less than 0) for other pages in the buffer. And the page with the smallest counter will be evict, whenever needed.
2. **Insertion, Deletion and Bonus**: In insertion and deletion, my implementation just follows the requirement in the document. For insertion, the B+ tree first call *findLeafPage* to find out the correct place for insertion, and *splitLeafPage* (this function may recursively call the *splitInternalPage*) will be called if the leaf page is full. The dirty pages will be returned and handled by the *BufferPool*, marking those pages and adding them into buffer; for deletion, the B+ tree will delete the tuple directly and redistribution and merging may be called to balance the tree. As for bonus, actually, no bonus has been mentioned in the document.

API changes

I changed no API in this lab, but added some helper functions to make the code more clear.

Missing Element

I implemented all elements required in this lab.

Timing and difficulties/confusing

It takes me about 14-15 hours to finish this project, including reading the documents, implementing all the required parts and debugging. As for difficulties or confusing, I found it difficult debugging the code, since the unit test is quite complex and it is a little hard to keep track of the data in the database. Also, I found that the *equals()* function may be missing for the *Tuple* class in the codebase, which may lead to failure of the test *testSplitRootPage* when the buffer size is set to 1 or using the MRU eviction policy.