DESIGN DECISIONS

- In the eviction strategies I implemented (LRU & MRU), I needed a data structure to store the pages which are suitable candidates for eviction, incase the bufferpool is full. Also, it was required to identify the usage of an eviction page to choose it as the least recently used or most recently used. To achieve this, I used a **linked list** which easily allowed me to add/remove pages. Whenever a new page is introduced to the bufferpool, I add it to the end of this linked list. This way, the front of my inked list gives the least recently used eviction candidate page and the end of the list gives the most recently eviction candidate page. I create this linked list as a BufferPool class attribute so that all the methods in this class can use it.

CHANGES TO THE API

- All the required changes have been made in 2 files: BufferPool.java & HeapPage.java

- Initialize the buffer with null pages upto a size(numPages) in the Buffer pool constructor

- Implement the logic for retrieving a page(if requested by a transaction) in Buffer pool getPage(). This includes checking if a page exists in bufferpool, loading it in bufferpool if not found there, updating appropriate variables (pin count, numhits, nummisses) to track eviction candidates & record page hits & misses in bufferpool, and returning appropriate exceptions if buffer is full or no eviction candidates present. All this is implemented in getPage(), pinPage(), unpinPage(), evictPage().

- Implement logic for getting number of empty slots in a page, check if a tuple if empty by checking it's slot bit in the header of a page and setting the header slot bit for a tuple. This logic is implemented in getNumEmptySlots(), getSlot() and setSlot().

MISSING/INCOMPLETE ELEMENTS IN MY CODE:

- NONE: All test cases have passed successfully with results matching with the results given on Sakai.

TIME SPENT:

- 5 days

DIFFICULT/TIME-CONSUMING PARTS OF THE ASSIGNMENT:

- The language of the assignment instructions is not very clear. It took me more time to understand the instructions than to write the actual code.