I. Experimental Environment

- Ubuntu 22.04
- JDK 1.8
- Ant 1.10.12

II. Experimental Procedure

Exercise 1: Page Eviction for SimpleDB Buffer

We chose LRU (Least Recently Used) as the eviction strategy, where the least recently used pages will be replaced. Before acquiring a page or after modifying a tuple, excess pages are reclaimed and relevant pages are placed into the buffer pool.

Exercise 2: Search in B+ Tree

Follow the standard algorithm specifically. Initially, I misunderstood the readme and thought I only needed to implement findLeafPage, so I kept failing and couldn't find the problem after many changes. Later, I realized that I also needed to complete Predicate and IndexPredicate. Also, do not forget to check if the input values are null.

Exercise 3: Splitting Pages

Splitting a leaf page requires modifying both the parent and sibling pages. When creating a new leaf page, it becomes the old leaf page's right sibling.

Exercise 4: Redistributing Pages & Exercise 5: Merging Pages

Our code and the code related to shape are very similar. Both consist of three steps: pulling down a parent entry, moving entries from another entry, and updating the parent entry. Operations on leaf pages always need to update the affected siblings, whereas actions on internal pages do not, as internal pages do not have siblings.

III. Time used

It took a long time to write this code. There were tests that didn't pass, but I didn't know what was wrong. Eventually, I realized that I might have accidentally modified some of the provided functions during debugging, which caused the test to fail. Therefore, I redownloaded the code, rewrote it, and finally passed.