PP3 B+ Tree Index Manager design document

For insertEntry:

We used a couple helper methods namely insertLeaf and insertNonLeaf

In insertLeaf we first check if the slot in the array we are looking at is open or not. If it is an empty slot then we insert and exit. If the slot is not empty then we keep searching for an empty slot. After we insert we check if the leaf is full and if it is full then we split. I`nsertNonLeaf is similar logic but we are checking if the node is a leaf or not. If not then we search the internal array for a node. In all the methods including insertEntry we made sure to only keep the necessary pages pinned. So as we went, we would unpin the page as soon as we were finished with it

For the scan Methods:

We used a helper method called findPage to set the currentPageNum to the page containing the low op value. In this method again we unpin the page matching the page Id immediately after we are done using it and before the next page is retrieved. so we don't have to worry about it in the destructor. In startScan we initially do some error checking to make sure none of the requirements are wrong before proceeding with the method. In scanNext as well we do error checking to essentially make sure that a scan is actually running before calling this method. As said before, we unpin the pages as soon as possible.

For Testing:

In main.cpp we wrote additional tests.

Test 4:

Test 5:

Test 6:

Test 7: