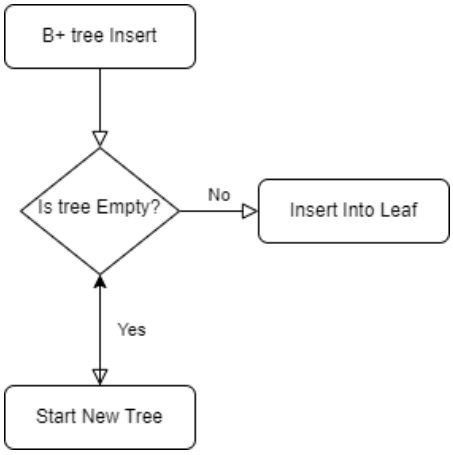
Lab 2 Diagram

Jahoon Koo

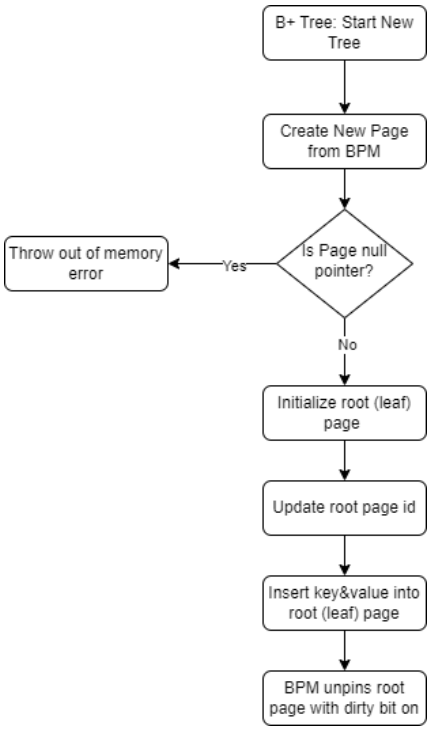
CSCI 5817

B+ Tree: Insert



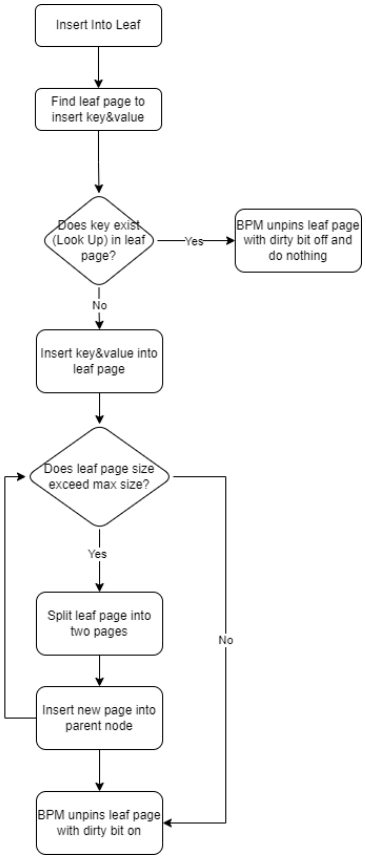
B+ tree inserts a key-value pair into the tree. It first determines whether the tree is empty. If the tree is empty, it will begin a new tree. Otherwise, it inserts a key-value pair into the leaf.

B+ Tree: Start New Tree



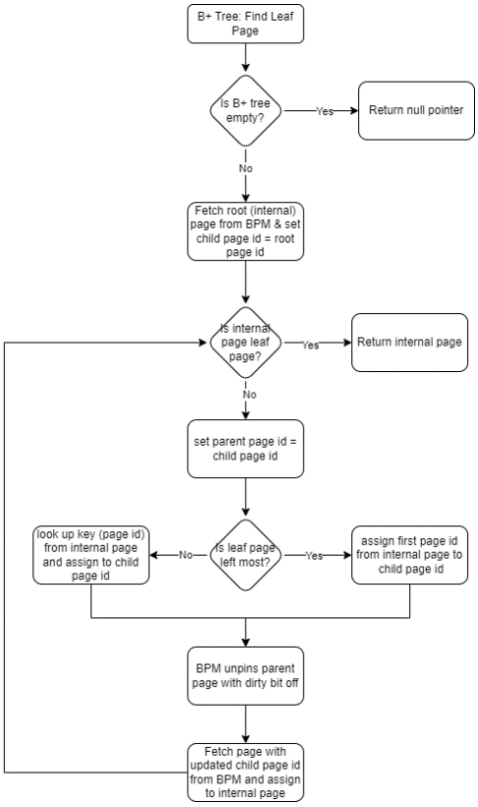
Start When the B+ tree is empty, the New Tree function is called. It first creates a new root page and obtains the root page id from buffer pool manager. Itt Throw out of memory error If the new root page is a null pointer due to a full buffer pool manager. If the new root page is valid (as opposed to a null pointer), create a leaf page with the root page id and update the root page id in the header page. When you first create a B+ tree, the root page is also a leaf page. The input key and value are then inserted into the root (leaf) page. The KeyIndex function is used during the leaf page insert to find the first index that is greater or equal to than the given key. When the correct key index is found, it shifts everything in the mapping type array, which contains keys and values in the leaf page, to the right of the key index, inserts the new key and value into the array, and increases the leaf page size by 1. Finally, buffer pool manager unpins the root page with the dirty bit enabled.

B+ Tree: Insert into leaf



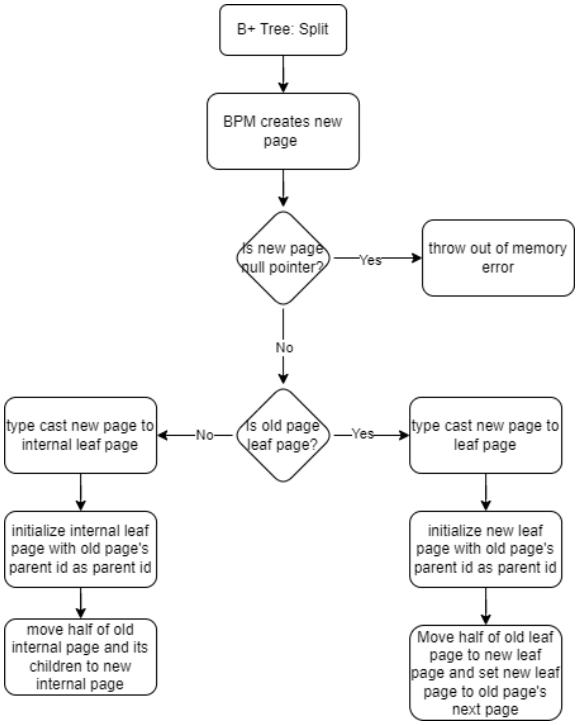
When the B+ Tree is not empty, the insert into leaf function is called. First, it finds a leaf page containing the specific key and determines whether the key already exists on the leaf page. If the key already exists in the leaf page, buffer pool manager unpins the page with the dirty bit set to false and does nothing because duplicated keys are not allowed. Insert key and value into the leaf page once there are no duplicated keys. If the leaf page size is greater than the maximum page size after insertion, the page must be split into two pages and a new page inserted into the leaf page's parent page. This procedure is repeated until the size of the leaf page is less or equal to the maximum page size. Finally, buffer pool manager unpins the leaf page with the dirty bit on.

B+ Tree: Find Leaf Page



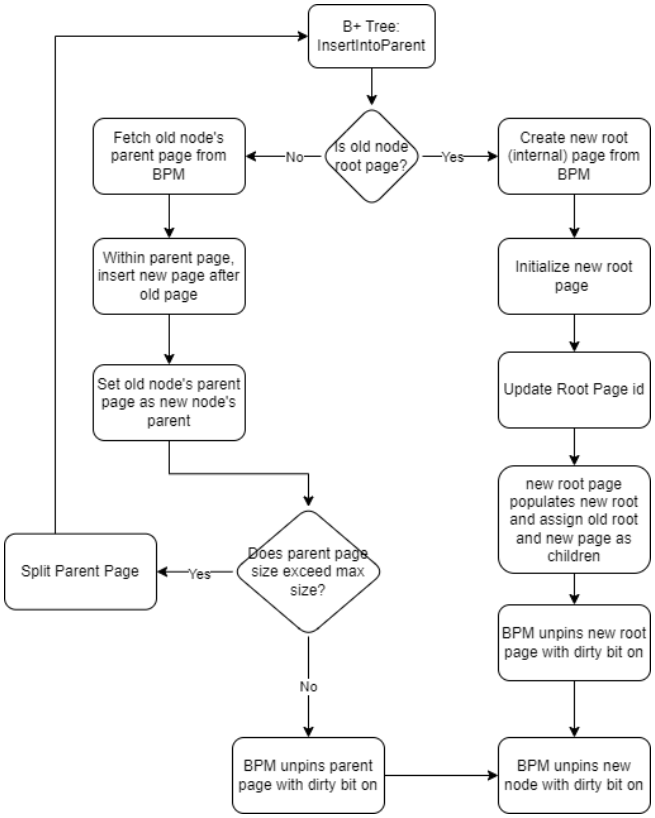
The Find leaf page function identifies a leaf page that contains the specified key. It first determines whether the B+ tree is empty and returns a null pointer if it is empty. Otherwise, it retrieves the root page from buffer pool manager using the root page id. It traverses the tree's internal pages. If the leftmost flag is set, proceed to the leftmost child page. Otherwise, it looks up child page id, which points to the child page containing the input key from the current internal page, and proceeds to the child page. Before proceeding to the child page, buffer pool manager unpins the parent page (current internal page) with the dirty bit off. The traverse is completed when it reaches a leaf page.

B+ Tree: Split



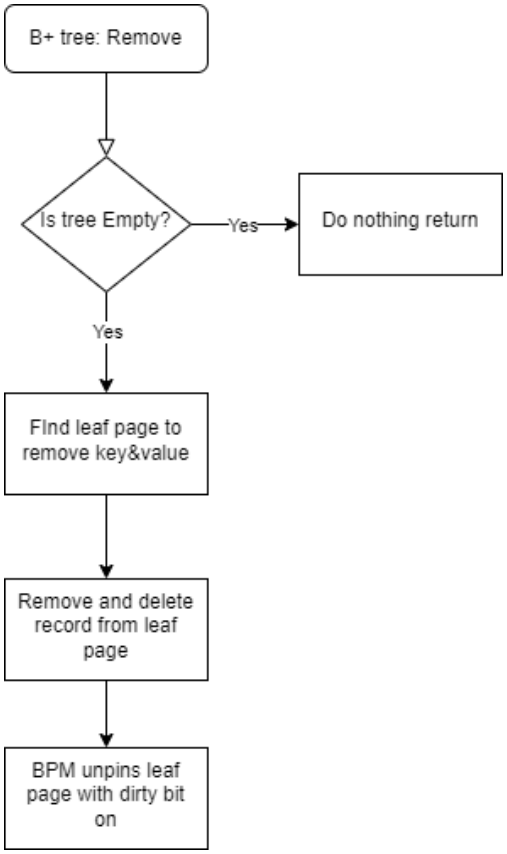
The Split function divides the input page and returns the newly created page. First, it generates a new page from buffer pool manager and checks to see if it is a null pointer. It typecasts the new page based on the input page's type. It initializes a new leaf page and sets the parent id of the input page to the parent id of the new leaf page. It moves half of the input leaf page to the new leaf page and reduces the size of the input leaf page by half, and the new leaf page becomes the next page of the input page. If the input page is an internal page, it initializes a new internal page and sets the parent id of the input page as the parent id of the new internal page. Finally, it copies half of the input internal page and its children to the new internal page and resizes them.

B+ Tree: Insert into parent



After splitting, insert into parent function inserts key and value pair into internal page. If the current page is root, it retrieves root page id the buffer pool manager's root page id and creates a new root page. It updates the root page id in the header page and populates a new root page in the root page by assigning the current page's id as the first key and the new page's id as the second key. Then, both the current page and the new page make the new root page their parent page. Finally, buffer pool manager unpins the new root page and the new page with the dirty bit enabled. If the current page is not the root, it retrieves the current page's parent page from buffer pool manager. inserts a new page after the current page within the parent page. The current page's parent page then becomes the parent page of the new node. If the parent page size exceeds the maximum size after insertion, it splits the parent page and inserts a new page from the split into the parent. Split and insert into parent repeats until the parent page is no longer larger than the maximum size.

B+ Tree: Remove



The Remove function deletes the key and value pair associated with the input key from the B+ tree. If the current tree is empty, nothing is done and the program returns. Otherwise, it selects the appropriate leaf page as the deletion target. Within the leaf page, it finds the index where the input key is located and shifts everything in the mapping type array to the left of the key index over, reducing the leaf page size by one. Finally, buffer pool manager unpins the leaf page with the dirty bit enabled.