Assignment 8

Fu Guanshujie, Ge Yuhao, Lou Haina, Qiu Xiaomin

April 2021

1

1.1

Start from the root, to find the maximum key, using the loop, and use the index of n(the number of keys in the node)get the rightmost pointer in the array and get the rightmost pointer of this pointer until it reaches the leave node. The rightmost value of leave node is the maximum key. Start from the root, to find the minimum key, then use index 0 to find the leftmost pointer and loop until find the leave node. The leftmost value (index 0) of leave node is the minimum key.

1.2

To find predecessor of the a given key. First we need to find the Btree node which store the given key. Using recursive function "search", the return value is a pointer a node which has given key. Then use loop to find the index of the given key in the node. 1. If the key is not on the leave, start from the right subtree of the given key(the pointer point to key values larger than given key), and loop the left most pointer until leave node to find the minimum value in the right subtree. 2. If the key is on the leave but index is not 0. Just find key of the index-1 stored on the same node. 3. If the key is on the leave also index is 0, we need to search the parents. We can start from the root and compare the value smaller than the given value and find the closest value and return it. The process of finding successor is similar as the finding predecessor, the only difference is The index is the number of keys stored(it is the rightmost key stored in the leave node), we need also search from root to find the closest bigger value with the given value.

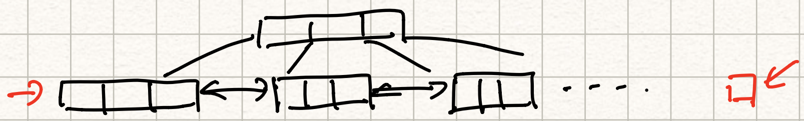
See the code in attached files.

2

See the code and answers in attached files.

Exercise 2

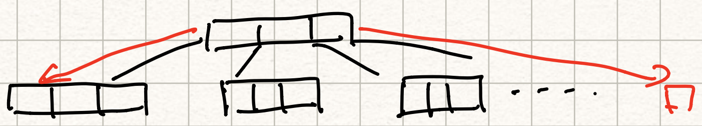
(i) if we have a chain between all the leaf nodes in the structure



Only need to find the head of the chain to find the min

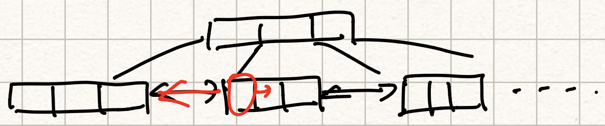
and find the end of the chain to find the max

if we don't have a chain between all the leaf nodes in the structure



need to go from the root and keep using the left most pointer and index 0 to find the min. and go from the root and keep using the right most pointer and index max to find the max

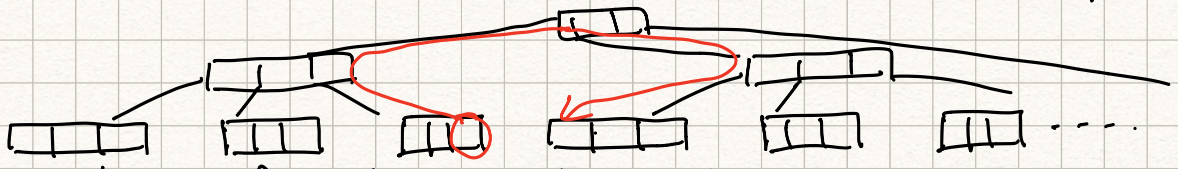
(iii) if we have a chain between all the leaf nodes in the structure



Only need to find the left pointer of the node to find the predecessor

and find the right pointer of the node to find the successor

if we don't have a chain between all the leaf nodes in the structure



need to go from the parent of the node and go to the right cousin of this node. then keep using the left most pointer and index 0 to find the successor, if there is no right cousin, go to the upper layer

The way to find the predecessor is similiar