**B-Tree**

Program description

김윤기

**Node Class**

This object includes just keys, pointers, data (value), r variable.

**B-Tree Class**

Parameter t is maximum degree.

The search function use the binary search because each node’s key are sorted.

The insert\_key function works recursively to check each node and split.

When the number of node’s key is greater than t, split function works. This function makes two new node.

The delete\_key function works recursively to rotate and merge. Once the rotation is done, the deletion process is finished because each node has a key of at least minimum degree. After merge, if the parent node has a key less than minimum degree, it checks the parent nodes in turn and proceeds with rotation and merge to ensure that all nodes have a key of at least minimum degree.

The find\_siblings function finds the node that has the key. And it returns [left sibling, right sibling, index of one’s own, parent node].

The rotate function performs left-to-right rotation, right-left-rotation, depending on the leftside value.

The merge function performs a merge with the left sibling, the right sibling depending on the leftside value.

**Test Function**

The insertion\_test function loads the input data and makes B-Tree. And then, the insert\_key function works with input data. After the insertion step, it prints out “Finish the insertion step”. Also, it stores the results of the search function in the result variable. The result variable is saved as a file in the “insetion\_output#.csv”. After search step, it prints out “Finish the search step”. Finally, it compares result variable and input data. If result variable is same as input data, it prints out “The result is the same as the input”.

The deletion\_test function loads the input data, delete data and delete\_result data. Then, it makes B-Tree and performs the insert\_key function with the input data. After the insertion step, it prints out “Finish the insertion step”. Also, it performs the delete\_key function with the delete data’s key. After the deletion step, it prints out “Finish the deletion step”. Also, it performs the search function with delete\_result data’s key. If the key is in the tree, it stores the corresponding value in the result variable. Otherwise, it records NaN in the result variable. The result variable is saved as as a file in the “deletion\_output#.csv”. Finally, it compares result variable and delete\_result data. If result variable is same as delete\_result data, it prints out “The result is the same as the output”.