COP5536 Fall'19 Project

Name- Aakriti Singh
UFID - 11399881
aakritisingh@ufl.edu

This project is implemented in java. Two data structures are used in implementing this project that are Red black tree and min heap. The structure of the project and method prototypes are described below.

1. Node.java- This class implements the building node elements. There are 12 attributes of a node that are degree of node in RB tree, parent of node inRB tree, left child of node in RB tree, right child of node in RB tree, color of node in RB tree, degree of node in min heap, parent of node in min heap, children of node in min heap, building number, execution time, and total time. Following are the functions used in this class.

a. getRightChild() - Gets right child of a node in RB tree.
b. getLeftChild() - Gets left child of a node in RB tree.
c. getColor() - Gets color of a node in RB tree.
d. getDegreeRB() - Gets degree of a node in RB tree.
e. getDegree() - Gets degree of a node in min heap.
f. getParent() - Gets parent of a node in min heap.
g. getRightSibling() - Gets the right sibling of a node in min heap.
h. getChild() - Gets children of a node in min heap.
i. getBuildNum() - Gets Building num of a node.
j. getExecTime() - Gets Execution time of a building/node.
k. getTotalTime() - Gets Total time of a building/node.
l. setDegree(int) - Sets degree of a node in min heap.
m. setRightSibling(Node) - Sets the right sibling of a node in min heap.
n. setParent(Node) - Sets parent of a node in min heap.
o. setChild(ArrayList<Node>) - Sets children of a node in min heap.
p. addChild(Node) - Adds child to a node in min heap.
q. setBuildNum(int) - Sets Building num of a node.
r. setExecTime(int) - Sets Execution time of a building/node.
s. setTotalTime(int) - Sets Total time of a building/node.
t. increaseDegreeRB(int) - Increase the degree of a node in RB tree.
u. setDegreeRB(int) - Sets the degree of a node in RB tree.
v. setParentRB() - Sets parent of a node in RB tree.
w. setRightChild() - Sets right child of a node in RB tree.
x. setLeftChild() - Sets left child of a node in RB tree.

y. setColor() - Sets color of a node in RB tree.
z. getParentRB() - Gets parent of a node in RB tree.

2. DataStruct.java- This class implements the insertion and deletion of a building in min heap and red black tree according to the input and requirement of the project. It also works on a building according to the input. Following are the functions used in this class-

a. insert(int, int , int) - Inserts a node to the min heap and RB tree.
b. formRBT(Node) - This function is called by the insert function. It inserts the node into the RB tree. This method calls the traverse method.
c. traverse(Node,Node) - This function finds the place to insert the new node. This function calls the balance function.
d. balance(Node) - This function balances the RB tree after insertion of a new node. This method calls the recolor, leftRightRot, rightRightRot, rightLeftRot, leftLeftRot as per different cases.
e. recolor(Node) - This function changes the color of a node in RB tree if required.
f. leftRightRot(Node) - This function performs a left-right rotation in the RB tree to balance it.
g. rightRightRot(Node) - This function performs a right-right rotation in the RB tree to balance it.
h. rightLeftRot(Node) - This function performs a right-left rotation in the RB tree to balance it.
i. leftLeftRot(Node) - This function performs a left-left rotation in the RB tree to balance it.
j. operate(int, int, int, int) - This function works on a minimum building for 5 days or for required time. And deletes the building if completed. This method calls the delete, changeET, search methods.
k. changeET(int time, Node node) - This method increase the execution time of a building. It also call the heapify and changeMin method.
l. changeMin() - This method changes the min node after heapify.
m. heapify(Node) - This method heapifies the min heap after change of execution time. It also calls the swap function.
n. swap(Node,Node) - This method swaps two nodes mainly a parent and child to heapify the min heap.
o. delete(Node) - This method deletes a node from the minHeap. It also calls the deleteRB and mergeInit method.
p. mergeInit(int, Node[])- This is the initiator method for the merge method.
q. merge(Node, Node)- This method merges two trees of a min heap. It also calls the linkSibling method.
r. linkSibling(Node,ArrayList<Node>)-This method links all the sibling to its right sibling in the min heap.

s. deleteRB(Node) - This method delete a node from the RB tree. It also calls the rebalancing method.
t. rebalancing(Node, String) - This method rebalance the RB tree after deletion. It also calls method rotateLR, rotateRL and itself according to different cases.
u. rotateLR(Node) - This performs a left right rotation in a RB tree after deletion.
v. rotateRL(Node) - This performs a right left rotation in a RB tree after deletion.
w. search(int,int) - This method is the initiator for the method searchX. This method also calls the method searchX.
x. searchX(Node,int,int,int)- This method finds all the building between building num 1 and building num 2.