COP-5536
Fall 2019
(Advance Data Structures)

Programming Assignment 1

Building Construction for Wayne Enterprises(Rising City project) using the concept of Min heap and red Black trees using unique execution time as unique key for min heap and building ID as unique key for RBT. Programming done in java.

LAVISH MEHTA

UFID:7981-8557

lavishmehta@ufl.edu

Compilation and running instructions:

To implement the project, I have implemented four base classes in the project respectively:

- **risingCity.java (main class)**
- **building.java**
- **RedBlackTreeimp.java**
- **MinHeapImpl.java**
- input.txt

Step1: The following file can be compiled by simply using the **make** command.

Step2: run the main class i.e. risingCity.java:

By using sys- **java risingCity <input_file_name>**

>input file having name "input.txt" is already placed in the zipped file. One can replace the input instructions or can simply place any other file by making sure the correct filename is specified while running the main class.

Environment:

The provided code can be tested on UNIX/Linux systems and windows.

Classes and Methods prototypes:

- **Class risingCity.java**

Main function is contained in this class, and the execution starts with this class only. It reads input from a file and process it according to the given command. It sets the condition and calls the object of both RBT and MinHeap classes. The buildings for Wayne enterprises are build according to the conditions defined here.

Some of the important methods:

Main function() – reads input from the file and evoke the method to start the scheduling of the buildings to construct. It starts following function:

1. **construction function() :** It starts the scheduler function and process the inputs according to the instructions. It starts the construction of building parsing line by line the input into it. It sets certain parameters to check conditions like execution time, total days to build the building, has the building is completed. It uses certain others methods of the class also.

2. **execute_tree()** : This is the main method that performs the logic part for the Wayne enterprises. Functions like insertion, deletion to Red black tree and Minimum Heap are performed here. Some operations like create building and insert are also called according to the conditions:

3. **CreateBuilding():** Will create a new building from insert() to the scheduler and will start executing it and then call method insertBuilding()**.** It will add new to the scheduler which will be added to the RBT and Min-Heap and will start the execution. It will not insert any other building before the completion of first 5 days of the current building.

Class: Building

This class is responsible for processing and returning the various values requested by risingCity.java class. There are functions like Java get and set methods and which is responsible to return corresponding values. Methods like getTotaltime() , getBuildingId() , getExecutionTime and many more functions to set the values.

Class: RedBlackTreeImp.java

This class implements the red black tree with all the necessary functions. All the Building ID's and the **execution time** of unique BuildingID stored in this tree. Methods:

insert() – insert a new job in the tree which has a unique Building_ID,

translate() – change the parent of a node if altered.

delete() - to delete a node based on Red Black tree mechanism.

Rotate left() and Rotate right()- these are used to have the corresponding rotation on insertion and deletion.

Node search() – A function to search weather the given node is in the left or right child in the tree .

Repair()- This function is use to fix the RBT after any operation performed on it.

Class: MinHeapImpl.java

This class implements of the minimum heap tree. It stores the executed time for which each building has been constructed or executed.

Methods:

insert() – insert into Min Heap based on the execution time sorting.

minHeapify()- send the smallest element to root and replace the current element based on Minheap translation and reconstruct into a min heap after any operation.

extractMin()- return the minimum element.

swap()- It is used to return the swapped nodes whenever we need to set a Min heap according to its translation.