PROJECT REPORT

Name: Shubham Saoji
Ufid: 26364957
Email: shubham.saoji@ufl.edu
Topic: COP 5536 Fall 2019

This project aims to develop software for Wayne construction which is developing a new city and has the below requirements:

- Wayne Construction works on only one building at a time.
- New building to be worked on should be selected such that it has minimum executed time.
- To print building number, executed time, total time either for a single building or building numbers in a given range depending on the input provided.

Design:
We have used two data structures, Red Black Tree and Min Heap. Red Black Tree stores nodes containing attributes building number, executed time, total time, left child, right child, and parent. The nodes are placed in RBT according to the building number.

We need to select and work on the building whose executed time is minimum. We use Min Heap to store nodes with attributes building number, executed time, total time and reference to corresponding RedBlackTree node. This reference is because if we need to delete an entry from min-heap, we can delete the corresponding node from

Working of RBT:
Whenever a new building is inserted in RBT, we call the insert method of RedBlackTree class and initialize node, place it in the tree as per its position based upon building number and adjust tree structure if there are two consecutive nodes of color Red. The attributes left child, right child, parent are set/adjusted accordingly.

Delete operation-
We categorize the node to be deleted as leaf node, degree 1 node or degree 2 node as per the degree of that node. If it is degree 2 node, we swap it with the node having the highest building number in the left subtree and then proceed for deletion. The RBT property may be violated after delete and rebalanceTreeOnDeletion ensures that this is fixed by color change of nodes or rotations or both

Working of Min Heap:
On insert of a new building, its executed time is 0 and node with input values is inserted in the last position of min heap which is implemented as an array. After insertion, we check if any rule of min-heap is violated that is if the child value of executed time is less than the corresponding parent value. In such a scenario, we adjust tree by swapping that child and its parent.

RemoveMin method of MinBinaryHeap class returns the node with minimum execution time. In case if there’s a tie in the executed time of two nodes, the node with minimum building number is given priority. The min node is replaced by the last node in min-heap and then the heap is checked for any violation and fixed if any.

Working of the project:
The input is read from the input file. As soon as the global counter equals command execution time provided in input, we perform the operation specified. Operation is Insert(building_nbr, total_time) or PrintBuilding(building_nbr) or PrintBuilding(building_nbr1, building_nbr2).

In case of Insert, we need to insert data in our system. PrintBuilding (buildingNum) prints the triplet buildingNume,executed_time,total_time. PrintBuilding (buildingNum1, buildingNum2) prints all triplets bn, executed_tims, total_time for which buildingNum1 <= bn <= buildingNum2.

I have assumed that a new project is inserted in software at end of the day, which means we’ll start working on that building or consider that building for selection of building with least executed time from the next day. Example: If a building is inserted in the system on the day(global counter)= 0, we consider it in building selection from day = 1.

So, flow is as follows-
We first check if there’s any building we are already working on, if yes we increase the executed time of that building by 1, if there’s no currently worked upon the building, we select a new building to work on (removeMin operation on min-heap). After working on a building, we check if its complete(executed time = total time). If yes, we print the building number and day on which it was completed and remove the entry of this building number from both data structures. If it is still not complete, we insert it back in min Heap with updated values.

After this, we check if there’s any input command for that day, that is insert or printBuilding commands and execute these commands accordingly.

It is important to note that, printBuilding is executed after deletion of the node which means, if a building is completed on a particular day and there’s printBuilding command for that day, the building that got completed is not printed as it has already been deleted from data structures.