Wayne Enterprise

Advance Data Structure-COP5536

Jyotik Parikshya

UFID: 0577-1359

Jyotik.parikshya@ufl.edu

1.Basic components of the program

sNode

It is an object that represents a basic node for both RBT and MinHeap. Each Node represents a building to be processed. It has the following Components:

int buildingNum_key_RBT :

Building Number of the Building . This Field is used in both min Heap and Red Black Tree.

Int executed_time_key_MH:

Executed Time till the current Global Time. This Field is used in both min Heap and Red Black Tree.

Int total_time :

Total time needed for completion of building. This Field is used in both min Heap and Red Black Tree.

sNode parent:

Parent of the node. This Field is used in only Red Black Tree. This field is insignificant for Minheap.

sNode left :

Left child of the node. This Field is used in only Red Black Tree. This field is insignificant for Minheap.

sNode right:

Right child of the node. This Field is used in only Red Black Tree. This field is insignificant for Minheap

Color color :

Color of the node. In this case it van be either black or red. . This Field is used in only Red Black Tree. This field is insignificant for Minheap

Min Heap: Object to represent the Minheap

sNode[] Heap :

Heap is an array of objects which represents the Min Heap.

Int size:

Stores the size of the Min Heap.

RedBlackTree: Object to represent the Red Black Tree

sNode root :

Root of the red black tree. It is of type sNode which will contain all the left and right children.

risingCity: Object representing the whole city under process.

int gtime :

Stores the current global time

RedBlackTree rbt:

Corresponding Red black Tree object

MinHeap mh :

Corresponding Min Heap object

int isprocessing :

Flag for determining if a Node (Building) is under process or not.

int counter:

Counter which acts as a local timer. It resets every time after reaching 5.

Int issaturated:

Flag for determining if any building’s work has been completed in a day.

2. Basic Operations in the Program

a) MinHeap:

1) Returnmin:
   returns the Minimum value node with respect to execution time.
2) Insert:
   Inserts the Element in MinHeap and Heapifies it.
3) Remove:
   Removes the minimum valued node from Min Heap and then Heapifies it.
4) minHeapify:
   Complete Build Heap of the Min Heap.
5) Tempinsert:
   Inserts the element without heapify.

b) RedBlackTree:

1) add:
   Adds the given Node in the RBT and does the adjustment according to the building Number.
2) remove:
   Removes the given node and does adjustment according to the Building Number.

3) print(int buildingnumber):
   Prints the details of a single building
4) print(int buildingnumber1,int buildingnumber2) :
   Prints the details of multiple buildings within that range.

c) risingCity:

1) pickdailybuilding :
   This functions selects the building to be worked upon when the processing of previous building is finished.
2) Updateprocessing:
   This functions updates the node which is under processing and checks whether further processing is required and if the work on the building is completed.
3) Removeandprint:
   Removes the node(building) whose processing is completed and prints it.

3. Flow of the Algorithm

Algorithm Basically runs in two parts:

- First we read the File using a File Reader in Java. For each input line from text file, we extract the time and compare with global time. If it is equal to current global time it will perform the operation to be performed ,if not then it will go into another while and increment the global time value until it becomes equal to the time for the input line. While it is in this loop it will pick the lowest execution time building and update it each time global time is incremented.

- Once the input file is over, the control will move over to another while loop which will keep on updating the buildings following the same rule as before without taking care of the conditions.

IMPLEMENTATION:

Folowing are some assumptions:

1) Only one building can be processed at a time.
2) If one building is in process and an insert statement is encountered the node will be inserted in the heap but wont be considered till the current processing is over.
3) Print operations will be executed after that day’s processing.
4) Completed Buildings will be printed after all operations.

During Processing there can be 4 conditions:
(Local time means the counter value, which varies from 0 to 5, which represents the number of days the building currently being processed has been processed from the point it was chosen as the Min Value)

1) Local time is not yet 5 but the Building is saturated: reset counter and extractmin.
2) Local time is not yet 5 and Buildiing is not saturated: counter++ and executiontime++
3) Local time is 5 and building is not Saturted : reset counter and executiontime++
4) Local time is 5 and building is saturated: reset counter and extractmin.

Insertion has been handled in two ways:

1) Insert into Min Heap and Heapify→ return Node that is inserted→insert the same node in Red Black Tree
2) Insert into Min Heap→return Node that is inserted→insert the same node in Red Black Tree-> Build Heap After processing is complete.

For Deletion:

ExtractMin deletes and returns the deleted node from Min Heap and then the same node is passed into delete function of Red Black Tree which removes the node and adjusts the RBT.

It can be observed that the same node has been used for both Red Black Tree and Min Heap. This has been done in order to connect these two data structure and keep track of them.