

ADS Assignment

This report consists of the compiler used, Object Structure, flow of code and description of the methods in the Red-Black Tree Assignment.

Environment

The code is developed on **Linux Ubuntu version 15.10** using Sublime Text Editor. The code is in C++ language. The compiler used is **g++ (Ubuntu 5.2.1-22ubuntu2) 5.2.1 20151010**.

Object Structure:

The code is divided in 3 files linked with each other.

- 1) **main.cpp**: This file contains the main method and the parse Input method. The main method contains the code for reading commands one after another and perform specific tasks. The parse Input method reads the initial sorted array from the file and calls the initialize method of RB-Tree instance.
- 2) **Node.cpp**: This is a class for storing the event information. It forms the nodes of the Red-Black Tree. The members are:
Id: Unique Id of each event. Type: Long.
Count: Stores the count for each event. Type: Long.
Color: Stores the color (r-red, b-black) of the node. Type: char
Parent: Stores reference to the parent node. Type: Pointer to node.
Left: Stores reference to the left child node. Type: Pointer to node.
Right: Stores reference to the right child node. Type: Pointer to node.
- 3) **RedBlackTree.cpp**: This file contains the whole structure of nodes. The members of this class are root node and nil node. The root node stores the reference to the root element of the tree. The nil stores the reference to the special sentinel node with black color. This class contains its constructor, initialization methods, Binary Tree specific methods, Red-Black tree specific methods, and this assignment specific methods.

Apart from these files, the zip contains the 'Makefile'. To create the executable named 'bbst', run command 'make' and to clear the executable, run command 'make cl'. Read further program specific instructions in README File.

Code Flow

Main Method: The program starts with the main method creating a Red-Black Tree instance and calling the parseInput method. When the tree is initialized, it reads commands one after another from standard input, and calls the appropriate functions for that command. If we read quit command, the program ends.

ParseInput Method: The parseInput methods reads the initial sorted array from the file whose name is given as command line argument and calls the initialize method of the RedBlack tree instance.

Initialize Method: This routine on RedBlack Tree class, performs following tasks.

- a) Convert the sorted Array to balanced Binary search tree.
- b) Finds the maximum depth of this tree and attaches sentinel node appropriately.
- c) Colors the last level valid nodes as red.

The initialization takes place in $O(n)$.

Method descriptions

This section describes the function members of RedBlack Tree class.

Increase Method-

Input – Id, value

Output – Returns the final count.

Description:

The three parameters (increase id value) are read from the standard input. If we find the node with id as key, we increase the count by value and return new count. If we don't find the node, then we create a new node, insert it, call the InsertFixUp method and return value.

Reduce Method-

Input – Id, value

Output – Returns the final count.

Description:

The three parameters (reduce Id value) are read from the standard input. If we don't find the node with id as key, we return 0. Else we decrease the count of the node by value and if the count is still above 0 then we return the new count or else we delete the node, call the DeleteFixUp method and return 0.

Count Method-

Input – Id

Output – Returns the final count.

Description:

The two parameters (count Id) are read from the standard input. If we don't find the node with id as key, we return 0. Else we return the count of that node.

Inrange Method-

Input – Id1, Id2

Output – Returns the final count.

Description:

The three parameters (inrange Id1 Id2) are read from the standard input. This method internally uses the next method multiple times to achieve goals. We start from ID1 and call the next Method multiple times until we reach the ID2 and keep adding the count of returned nodes. At last we return this count.

Next Method-

Input – Id

Output – Returns the next node. Returns sentinel if next is absent.

Description:

The two parameters (next Id) are read from the standard input. We try to find the node with given Id. If we find the node then use that node as starting node, else we use the previous node as starting node. Then we use the successor method to find next node. If successor node is present we return that node else the sentinel node.

Previous Method-

Input – Id

Output – Returns the previous node. Returns sentinel if previous is absent.

Description:

The two parameters (previous Id) are read from the standard input. We try to find the node with given Id. If we find the node then use that node as starting node, else we use the next node as starting node. Then we use the predecessor method to find previous node. If predecessor node is present we return that node else the sentinel node.

Initialize Method-

Input - Array of pointers to Nodes, start index, end index.

Output – The root node.

Description –

This method uses the divide and conquer strategy to achieve its goals. It find the middle element, creates root of it and then recursively call the same method with each half two time. Thus it uses the left subarray for building the left subtree and the right subarray for building the right subtree. All nodes are black initially. Then it uses the maxDepth method to find the maximum depth of this tree and attaches sentinel node appropriately. Then uses the ColorNodes method to color the last level valid nodes as red. The initialization takes place in $O(n)$ time.

InsertFixUp, DeleteFixUp Method -

Input – rooted node

Output –

Description – Uses the tree rotation operations to fix the red-black tree property after node insert.

Transplant Method-

Input – Node u, Node v

Output – Node

Description – Replaces the u subtree with the v subtree.

treeMin Method-

Input – rooted node

Output - Node

Description – Finds the min element in the subtree and returns the node.

treeMax Method-

Input – rooted node

Output - Node

Description – Finds the max element in the subtree and returns the node.

There are a few more methods present for helping purposes, but are secondary and are not re-black tree or assignment related. They are intuitive and are provided with comments.

To run the program read the README file that's included.