Advanced Data Structures
COP 5536

Spring 2016

Project Report

Submitted By

Abhisek Mohanty
34931899
avisec@ufl.edu

Report Overview

1. Associated Files

#	File Name	Basic Outline
1	bbst.java	main class. Contains : • Logic for taking input from the command line as argument, in the form of a file and create the Red Black Tree from the input. • Logic for calling the respective methods based on the terminal input (commands) given by the user. Program stops when user inputs a “quit” command.
2	Event.java	The Event class is defined in this file. Has two properties. 1. Id 2. Count This class is referenced in other classes.
3	RedBlackTree.java	This class holds the implementation of the Red Black Tree and the 6 operations mentioned in the question. Methods in this class are outlined in section 3. The **RedBlackNode class** is defined here.
4	makefile	

2. Classes

#	Class Name	Primary Variables	Methods
1	Event	Id, Count	Constructor
2	bbst	-	main()
3	RedBlackNode	element color left right parent	Constructor
4	RedBlackTree	nil rightMostNode	Detailed in Section 3

3. Methods

a. RedBlackTree.java

#	Method Name	Description
1	insert(Event item)	This method is called from the main class to insert a new item into the Red Black Tree. Every time, a node is inserted according to the BST property, the RB_Insert_Fixup method is called to make sure that the Red Black Tree properties are not violated. Used the value of the rightMostNode property, to find the insert position in O(1) time. Hence, tree is initialized in O(n) time.
2	RB_Insert_Fixup(RedBlackNode node)	This method is called from the insert method. Parameter passed is the newly inserted node. Calls the leftRotate and rightRotate based on the parent being the left or right child of its parent.
3	leftRotate(RedBlackNode node)	Called from the RB_Insert_Fixup method. Used to left rotate the tree based on the conditions that are checked in its parent method RB_Insert_Fixup. This is also called when an element is deleted from the Red Black Tree.
4	rightRotate(RedBlackNode node)	Called from the RB_Insert_Fixup method again. Used to right rotate the tree based on the conditions that are checked in its parent method RB_Insert_Fixup. This is also called when an element is deleted from the Red Black Tree.
5	deleteNode(RedBlackNode z)	The deleteNode method is used to delete a node from the Red Black Tree. In the context of the program its only called from the reduce method when the Count corresponding to a given Id falls below 1. The deleteNode method has other helper methods that are defined below.
6	RB_transplant(RedBlackNode u, RedBlackNode v)	The Transplant method for tree deleteNode method. Used to replace one subtree as a child of its parent with another subtree.
7	RB_Delete_Fixup(RedBlackNode x)	Helper method for deleteNode to fix colors and RB Tree properties post delete is called.
8	treeMinimum(RedBlackNode subTreeRoot)	Called from the deleteNode method to find the minimum element of the tree rooted at subTreeRoot
9	findNodebyId(RedBlackNode root, int theID)	Given an Id, this method finds if a node corresponding to the given Id is present in the tree. If present, it returns the node, otherwise it returns null.

#	Method Name	Description
10	Increase(int theID, int m)	Increases the Count of the corresponding theID by m. Inserts a node with Id = theID and Count = m if theID is not present in the tree. Prints the Count of the given Id.
11	Reduce(int theID, int m)	Reduces the Count of the corresponding theID by m. If value of Count corresponding to theID falls below 1 after reduction, the deleteNode method is called to delete the node. Prints the Count of the given Id.
12	Count(int theID)	Prints the Count for the given Id. Calls findNodebyId first to check if the node is present. Prints 0 if the node is not present.
13	Inrange(RedBlackNode root, int id1, int id2)	Returns the total count for IDs between id1 and id2 inclusively.
14	Next(int theID)	Prints the ID and the count of the event with the lowest ID that is greater that theID. Prints “0 0”, if there is no next ID.
15	Previous(int theID)	Prints the ID and the count of the event with the greatest ID that is less that theID. Prints “0 0”, if there is no previous ID.
16	printCountforID(RedBlackNode node, int Id)	Prints the Count for the given node and Id.

b. bbst.java

#	Method Name	Description
1	main(string args[])	The main method which holds the logic for : - processing the input file passed as an argument - calling the insert method to create the Red Black Tree - processing the commands entered through the console

4. Program Structure, Compiler and Runtimes

Structure and Files	bbst.java - Main Method, start of the program. This class handles all the console related tasks and call appropriate methods in the other files. Event.java - Contains the definition of the Event Class. RedBlackTree.java - Contains the logic for the entire RedBlackTree. Major Methods related to : - Insert - Delete - InsertFixup - DeleteFixup - 6 Methods to be implemented in the project - Other Helper methods.
Compiled and Tested on :	OS X 10.11.3 javac -version = javac 1.8.0_73 on Java(TM) SE Runtime Environment (build 1.8.0_73-b02)
Running Times on test files :	test_10000000.txt - 12 sec with default heap size test_100000000.txt - approximately 3 mins with heap size 8000M