RED-BLACK TREES

Parth Hajari – 21BCE508
Jaimin R Babariya – 21BCE509
Prajit Matalia – 21BCE512

Guided By: –
Jitendra Bhatia

Introduction

Rudolf Bayer is a professor of Informatics at the Technical University of Munich where he had been employed since 1972. He is noted for inventing three data sorting structures: the B-tree (with Edward M. McCreight), the UB-tree (with Volker Markl) and the red–black tree.

A red-black tree is a kind of self-balancing binary search tree where each node has an extra bit, and that bit is often interpreted as the color (red or black). These colors are used to ensure that the tree remains balanced during insertions and deletions. Although the balance of the tree is not perfect, it is good enough to reduce the searching time and maintain it around O(log n) time, where n is the total number of elements in the tree.This is an improvement over the time complexities of Binary trees of O(N).

Introduction cont..

RB trees guarantee, in relation to other algorithms, optimal computational times for INSERT, DELETE and SEARCH operations. This fact allows their use in sensitive applications from the point of view of computation time, such as in real-time applications.
However, due to their characteristics, we can also use RB trees as fundamental building blocks in data structures underlying numerous applications.

- Most of the self-balancing BST library functions like map, multiset, and multimap in C++ (or java packages like java.util.TreeMap and java.util.TreeSet) use Red-Black Trees.
- It is used to implement CPU Scheduling Linux. Completely Fair Scheduler uses it.
- It is also used in the K-mean clustering algorithm in machine learning for reducing time complexity.
- Moreover, MySQL also uses the Red-Black tree for indexes on tables in order to reduce the searching and insertion time.

Structure

There are certain additional constraints that are followed in Red-Black trees. They are,

- Every node is either red or black in colour.
- The tree's root is always black.
- There are no two red nodes that are adjacent (A red node cannot have a red parent or red child).
- There is the same number of black nodes on every path from a node (including the root) to any of its descendants NULL nodes.
- All of the leaf nodes are black.
- The height of a red-black tree with n nodes is h<= 2 log2(n + 1).
- All leaves (NIL) are black.
- The black depth of a node is defined as the number of black nodes from the root to that node i.e the number of black ancestors.

Insertion

1. If tree is empty, create new node with black color
2. if tree is not empty, create newnode as leaf node with red color.
3. if parent of newnode is black then exit.
4. if parent of newnode is red,then check the color of parent's sibling of newnode.
    a. if the color is black or NULL then perform rotation and recoloring.
    b. if the color is red, then recolor & also check if parent's parent of newnode is not root node then recolor it & recheck.

Deletion

1. Perform the standard deletion operation of Binary search tree.
2. Find the minimum element on the right subtree or find the greatest element of left subtree.
3. Store the color of the node to be deleted.
4. Relace the element to be deleted with the selected element.
5. after deletion replace the element's color with the stored color.
6. Recolor the tree if all the necessary constraints are followed.
7. There can be three cases:
    a. Both the element's children are Null.
    b. Right child is Null.
    c. Left child is Null.

Thank You!