CSCI 311 Data Structures & Algorithms

LAB 2 Assignment

PROJECT DESIGN:- The project reuses almost all the code from of LAB2 assignment. The functionality is separated into different .cpp/.h files.

The project implements insertion in a red black tree. The file rbtree.h/.cpp has the declarations/definitions of Red Black Tree members. The file has 2 classes :

1. **Class Tree:** Includes the following:-

- **Insert function :-** void insert(pair<string,int>): implements insertion in Red black tree. Takes the pair<string,int> as its input and places it in the Red Black Tree .
- **Fixup:-** void fixup(Node*):The insertion violates the RedBlack Tree property hence fixup code implements the functionality to restore the same.
- **leftRotate:-** void leftRotate(Node*): Performs left rotation on the nodes to restore the red black tree property.
- **rightRotate:-** void rightRotate(Node*): Performs right rotation on the nodes to restore the red black tree property.
- **Inorder:-** void inorder(Node*): Performs tree traversal inorder.
- **Height:-** int height(Node*): Returns the black height of the tree.

2. **Class Node:-** Includes the following:
    - Data members: left child, right child, parent,color(to store the tree property).
    - Setter and getter methods for data members.

TIMING:-
The project uses a microsecond resolution timer to calculate the time required for insert function in Red Black tree and compares the same with the map implementation.

TIMING Observation:-
Red Black Tree time:-1.34601e+09 microseconds. Map time:-1.35232e+09 microseconds. Red Black tree is faster than stl map. STL map is implemented using Red Black tree data structure.

BLACK HEIGHT:- The black height of the tree for inputs 1,2,3,4,5 are 4, 2, 5, 8, 7, respectively.