LAB ASSIGNMENT-3

CSN-261

Rushiprasad Sagare

CSE- 2nd Year

18114071

Sub-batch O3

Problem 1-

Given the set of integers, write a C++ program to create a binary search tree (BST) and print all possible paths for it. You are not allowed to use subarray to print the paths. Convert the obtained BST into the corresponding AVL tree for the same input. AVL tree is a self-balancing binary search tree. In an AVL tree, the heights of the two child subtrees of any node differ by at most one; if at any time they differ by more than one, rebalancing is done to restore this property. Convert the obtained BST into the corresponding red-black tree for the same input. Red-Black Tree is a self-balancing Binary Search Tree (BST) where every node follows following rules.

1) Every node has a colour either red or black.
2) Root of tree is always black.
3) There are no two adjacent red nodes (A red node cannot have a red parent or red child).
4) Every path from a node (including root) to any of its descendant NULL node has the same number of black nodes.

Write a menu driven program as follows:

1. To insert a node in the BST and in the red-black tree

2. To create AVL tree from the inorder traversal of the BST

3. To print the inorder traversal of the BST/AVL/red-black tree

4. To display all the paths in the BST/AVL tree/red-black tree

5. To print the BST/AVL tree/red-black Tree in the terminal using level-wise indentation (print colour for red-black tree)

6. Exit

Data Structure Used- Binary Search Tree

AVL Tree

Red Black Tree

Algorithm-

1)Used cpp to program.

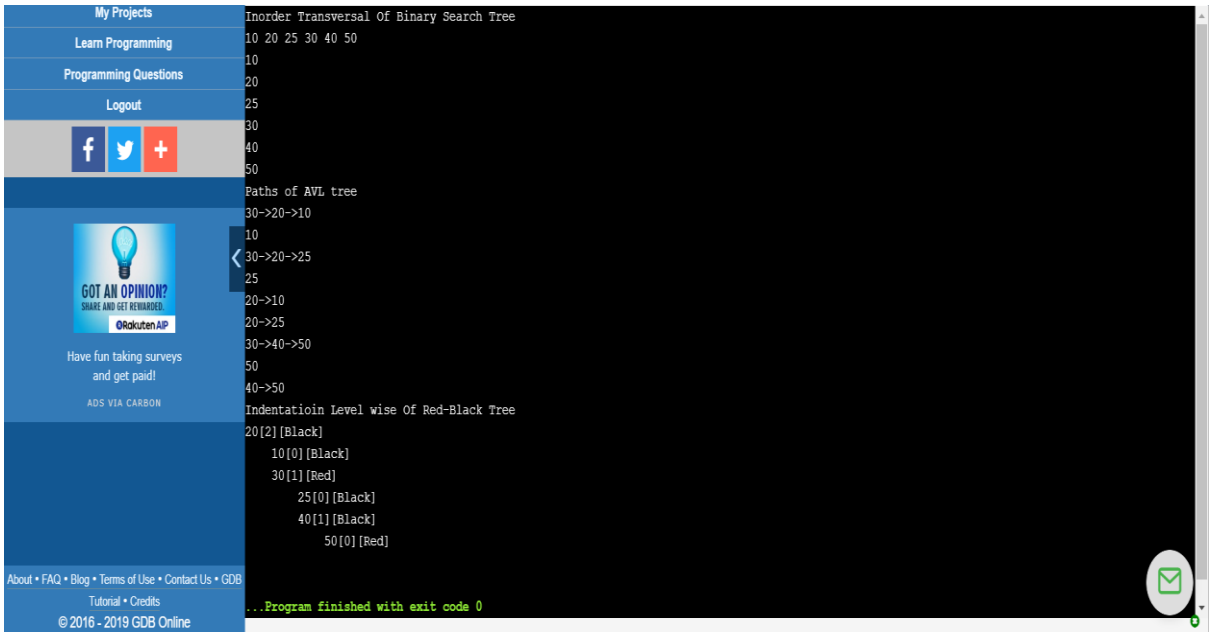
2) Used 3 separate classes for bst,avl and rbt.

3)Used struct node to make a node of the tree

4)Data structure used tree .

5)Used concepts of rotation to make red black. tree from given binary search tree .

6)Made avl tree by inorder traversal of the elements of binary search tree.



Problem 2-

For a given sequence of positive integers A1, A2, …, AN in decimal, find the triples (i, j, k), such that $1 \le i < j \le k \le N$ and $Ai \oplus Ai+1 \oplus \ldots \oplus Aj-1 = Aj \oplus Aj+1 \oplus \ldots \oplus Ak$, where $\oplus$ denotes bitwise XOR. This problem should be solved using dynamic programming approach and linked list data structures.

Input:

(a) Number of positive integers N.

(b) N space-separated integers A1, A2, …, AN.

Output: Print the number (count) of triples and list all the triplets in lexicographic order (each triplet in a new line).

Data Structures used- Linked List , Queue .

Algorithm-

1) The numbers are stored in a linked list.

2) A xorfunc is used to xor the elements of a linked list from start to end.
3) Triplets are counted using count function.
4) These triplets are printed using showTriplets function.
5) Used concept of dynamic programming
6) Used 2 pointers to traverse in linked list