CS593: Data structure and Database Lab
Take Home Assignment - 5 (10 Questions, 100 Points)

Submission Dead Line: 8-Oct-2021 23:59 Hours

Instructions:

1. Hard coding is not allowed in any questions. In case of hard coding assignment submission shall not be considered for evaluations.
2. Any manual entry of input to any program is not allowed. Your program should read all input from .txt file and write all output to other .txt file. You have to create your own input .txt file and output .txt file for each question. Submit input and output .txt file along with code.
3. Any copy case or plagiarism will be awarded (-100) marks as penalty. Hence please put your own effort to solve problems. Do not copy or share code with your friends. Submit code for only few questions is better than any copy code.
4. Any version or platforms issue is your responsibility. We have given all the required instructions and recommendations regarding compilers and versions. Please follow that. In case of any version issue your submission shall not be evaluated.
5. Your program should pass all test case. If your program fail for any test case 0 marks will be awarded for that question. Please don't write any mail to TA after evaluations to know the test case which your program fails. No explanations shall be provided regarding test case by TA before or after evaluations. You need to think and write your own test case. Make sure that your program passes all test case.
6. Please do not expect TA to remove first or last three line and then compile so that it run error free during evaluations. Why should I edit your submission? This is ethically wrong to edit student submission. How can a evaluator edit submission file? For any such request from student side (-20) marks will be awarded as penalty.
7. In case of your program have compile time error at our system during evaluation means it is error. There is no if and but. We can't run program at your pc for every student.

Problem Set:

1. Create a Binary Tree and write recursive function for following operations.
   - I. Find LCA (Least Common Ancestor) of two nodes.
   - II. Print all Ancestors of a node
   - III. Find vertical sum of Binary Tree.
   - IV. Convert a Binary Tree into its mirror.
2. Create two binary tree and write functions for following operations.
   - I. Given two binary trees, return true if they are structurally identical.
   - II. Given two binary trees, return true if the tree is isomorphic to each other and False otherwise.
3. Write a program to create a Binary Search Tree (BST) and write functions to performs following tasks.
   - I. Write non-recursive functions to find PreOrder, InOrder, PostOrder traversal of BST.
   - II. Write functions to find Inorder Predecessor and Inorder Successor of BST.
   - III. Write a function to delete a node from BST.
   - IV. Function to find K^{th} smallest element in BST.

V. Given a BST and two number K_1 and K_2, give an algorithm for printing all the elements in BST in the range K_1 and K_2

4. Write a program to construct Height Balanced Tree (HBT) given a set of number as input and perform following operations.

I. Write a function to insert element into HBT

II. Write a function to delete a node from HBT

III. Given a BST as input write a function to check whether it is an AVL tree or not?

5. Suppose we have arrays PREORDER[n], INORDER[n], and POSTORDER[n] that give the preorder, inorder, and postorder positions, respectively, of each node n of a tree. Describe an algorithm that tells whether node i is an ancestor of node j, for any pair of nodes i and j. Write a program to implement your algorithm.
6. Write a program to list the DATA fields of the nodes of a binary tree T by level. Within levels nodes are to be listed left to right.
7. Write an algorithm SWAPTREE(T) which takes a binary tree and swaps the left and right children of every node.
8. Write a program to construct the binary search tree with a given preorder and inorder sequence.
9. Write a recursive and non-recursive function that produces the height of a binary tree.
10. Description: Read the statement of problem G for the definitions concerning trees. In the following we define the basic terminology of heaps. A heap is a tree whose internal nodes have each assigned a priority (a number) such that the priority of each internal node is less than the priority of its parent. As a consequence, the root has the greatest priority in the tree, which is one of the reasons why heaps can be used for the implementation of priority queues and for sorting.

A binary tree in which each internal node has both a label and a priority, and which is both a binary search tree with respect to the labels and a heap with respect to the priorities, is called a treap.

Your task is, given a set of label-priority-pairs, with unique labels and unique priorities, to construct a treap containing this data.

Input

The input contains several test cases. Every test case starts with an integer n. You may assume that 1<=n<=50000. Then follow n pairs of strings and numbers l1/p1,...,ln/pn denoting the label and priority of each node. The strings are non-empty and composed of lower-case letters, and the numbers are non-negative integers. The last test case is followed by a zero.

Output

For each test case output on a single line a treap that contains the specified nodes. A treap is printed as (< left sub-treap >< label >/< priority >< right sub-treap >). The sub-treaps are printed recursively, and omitted if leafs.

Sample Input

7 a/7 b/6 c/5 d/4 e/3 f/2 g/1

7 a/1 b/2 c/3 d/4 e/5 f/6 g/7

7 a/3 b/6 c/4 d/7 e/2 f/5 g/1

0

Sample Output

(a/7(b/6(c/5(d/4(e/3(f/2(g/1)))))))

(((((((a/1)b/2)c/3)d/4)e/5)f/6)g/7)

(((a/3)b/6(c/4))d/7((e/2)f/5(g/1)))

Note: Any reference Height Balance Tree (HBT) is reference to AVL Tree.

Submission instruction

File Naming Convention: Create a directory with your roll number. Inside this directory, place all the programs and input files. Prefix the file name with your roll number followed by "_" followed by question number followed by ".c". Example: 194161000_q1. c.

README.txt Write a short note on sequence of steps involved to run your programs. Include what is the input for the program (with an example) and what will be the output from the program (with an example).

tar gzip Create (roll number). tar.gz file using the above directory. This directory must contain the above program.

Submission Email the above tar gzip file to the CS593 head TA **sujitkumar@iitg.ac.in** as per the above given dead line.

Submission does not follow above file naming convention and instruction shall not be evaluated.

Copying You should avoid indulging in copying. Every submission will be subject to software similarity using the **tool Measure of Software Similarity available at https://theory.stanford.edu/~aiken/moss/.** Do not copy code from any source including coding website. Two submissions having similarity score equal to or more than 10.0% will be declared copied. If you are found involved in copying act, your name will be referred to disciplinary committee. Therefore, you are requested to place individual efforts and avoid copying.