

CS218- Data Structures

Programming Assignment No. 3

Fall 2019

Instruction

This is the third programming assignment for the course CS218- Data Structures in the offering Fall 2019. The assignment comprises of two problems. It is suggested that you should start working on the assignment at your earliest. This seems a good amount of intellectual work required to complete it. Each question should be solved in one program file names as per suggested scheme. Your student number dash assignment number and problem number, e.g K182122-A3P1.cpp

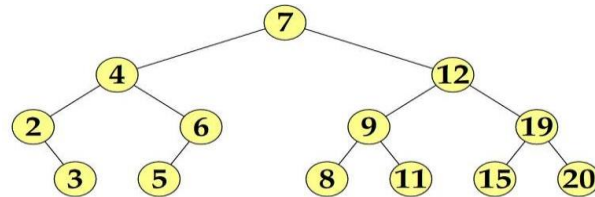
Your program should take input from the input file as per the direction of the input format. The output should be produced in an output file as per the required format. These two files should be read from the current directly where your source code is residing.

The assignment is for individual and there should not be any case of cheating. You can have discussion about any problem and approach among yourself but do not share code and instruction for any problem.

Due Date: December 01, 2019 21:00PM

Problem 1: Longest Order Sequences of BST Nodes

In a Binary Search Tree (BST), you need to find the longest increasing sequence of nodes (contains + values)- This is also a longest path that contains increasing or decreasing node values. You need to read a sequence of values from a file, this is the given order for which you need to insert these values into an empty BST. Consider the sequence 7,4,12,9,6,5,2,3,19,15,20,8, and 11



The above tree contains all the values and it is a BST. You need to find the longest sequence of node that is increasing or decreasing of the node values. In the above tree. 7, 12, 19 and 20 is the largest such node sequences forming path 7->12->19 -> 20

The main requirement for this assignment is to write your own BST code and utility functions that make the task easy for you to decide.

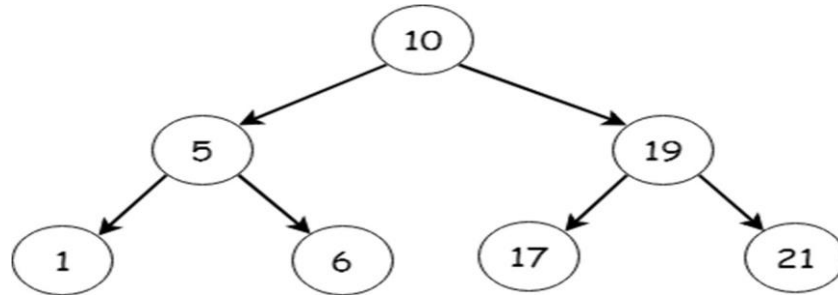
Input file format: The input file contains values for each node in a given order. You need to insert each value in an empty BST. The tree thus formed is used to find the longest sequence of increasing or decreasing values. A sample input is given below:

Output file format: The output file contains length of longest ordering sequence from BST, and second line contains node values comma separated.

Input File	Output File
7 4 12 9 6 5 2 3 19 15 20 8 11	4 7,12,19,20

Problem 2: Subtree of a Binary Tree with Conditions

You are given a sequence of distinct positive numbers that you need to insert into an empty Binary Tree (following the order of insertion that we discussed for it in the class i.e. from left to right). In the tree thus constructed you need to find any subtree which is a valid Binary Search Tree (BST) and the sum of all the node values of this BST equal to X. Consider the following Binary Tree



You can easily identify that a sub-tree from node 17 19 21 form a valid BST whose node sum is 57.

Input file format: The first line of the input file contains number of nodes in Binary Tree (say n), the second line contains the sum required to produce for any valid BST (say s). From the next line you will get values of each n nodes. You need to insert each value in an empty Binary Tree (BT). The tree thus formed is used to find a subtree which is a valid Binary Search and the sum of all the nodes included in the BST sums to X. A sample input is given below:

Output file format: The output file contains node values which sums to X.

Input File	Output File
7	17
57	19
10	21
5	
19	
1	
6	
17	
21	