

CS515: Computer System Lab 2

Date: 17th Feb 2022

MidSem

Submission Filename: `midsem.c` or `midsem.cpp` and `midsemREADME.txt`

Full Marks 100

Duration: 3 hours

Task 1: Creating the tree from the input data file

A binary search tree (BST) is to be constructed from the input (`ip.txt`) data file. Read the file, and create the BST using a pointer-based representation. Maintain links for left and right children. The nodes of the tree are specified in level wise manner. Thus, root node and its children are mentioned first. In the subsequent lines, the child nodes are specified in left to right manner. All the key value of the node are unique and positive in nature. For each node, its key and the corresponding left and right child key values are provided. If left or right child is not available then it will be marked by -1.

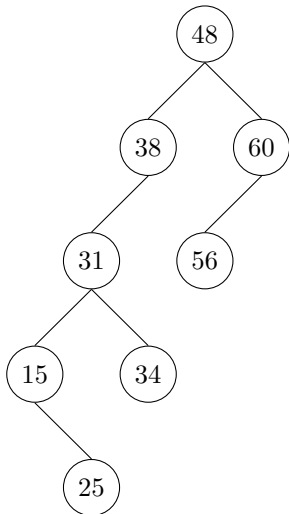
After creating the tree, it is required to return the pointer to the root node. In all of the subsequent parts, this pointer should be passed as the tree, not the raw data from the file.

Let's consider the contents of the sample input file is as below.

Filename: `ip.txt`

```
48 38 60
38 31 -1
60 56 -1
31 15 34
56 -1 -1
15 -1 25
34 -1 -1
25 -1 -1
```

The BST corresponding to this input file is as given below



After constructing the tree from the given input, check whether it is a valid BST or not. Also, print the preorder and inorder traversal of the constructed tree.

In this case,

the preorder traversal of the tree is 48 38 31 15 25 34 60 56

the inorder traversal of the tree is 15 25 31 34 38 48 56 60

Marks 40

Task 2: Finding the farthest leaf node(s) of the tree

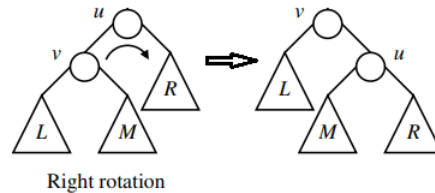
In this task, you need to find the farthest leaf node(s) from the root in terms of number edges required to reach from the root to the leaf node.

For the aforementioned example, the farthest leaf node(s): 25
The distance of the leaf from the root is: 4

Marks 30

Task 3: Finding the tree after performing right rotation

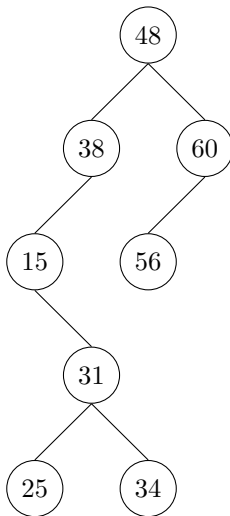
Use the right rotation as shown in the following figure to perform rotation w.r.t. a node. Each rotation changes the root of the subtree at the position of rotation (the root changes from u to v for right rotation, according to the figure).



For performing the right rotation, we need to enter key value of the node. If right rotation can be performed then it performs the right rotation without violating BST property and prints the preorder and inorder traversal of the tree.

For the aforementioned example,
Enter key value for right rotation: 15
Right rotation w.r.t 15 is not possible

Enter key value for right rotation: 31
Right rotation w.r.t 31 is possible
The tree after right rotation



the preorder traversal of the tree is 48 38 15 31 25 34 60 56
the inorder traversal of the tree is 15 25 31 34 38 48 56 60

Marks 30

Submission

Mention any important points of your strategy and any other assumptions that you have used in the `midsemREADME.txt` file. Report the time complexity of your strategy with necessary justification. Also, mention if you have any specific restriction on your input and output. Submit the program using the following submission link only.

https://www.iitp.ac.in/~samrat/CompSysLab2_CS515/submission/

Guidelines

- a Do not use any library/package (eg. STL etc) to implement this. Your code must be well documented and any invalid input must also be handled properly.
- b We will evaluate the codes using `gcc` or `g++` commands at the terminal mode.
- c There will be penalty if you are found to take any unfair means during the lab hours and during the assignment submission process.
- d Copying others' program and allowing others to copy your program will be equally penalized.