

# CS515: Computer System Lab 2

Date: 27th Jan 2022

Assignment 4

Submission Filename: `assign4.c` or `assign4.cpp` `assign4README.txt`

Due Date: 30th Jan 2021 9:00 am

## 1 Problem Description

In this assignment, you need to work with binary trees. Let a binary tree is denoted by  $T_{bt}$  with each node storing a key and two child pointers (L and R). Assume that the keys stored at the nodes are distinct from one another. Let  $n$  denotes the number of nodes in  $T_{bt}$ , and  $ht(v)$  indicates the height of subtree with root node  $v$ . From this tree you need to create a *left tilted binary tree* or  $T_{ltt}$  and also you need to print the trees in a specific format. The details of the different tasks are described next.

### 1.1 Construction of binary tree from the input file

The binary tree  $T_{bt}$  is to be constructed from the input file `ip.txt`. The user specifies each node by a triple  $(k, l, r)$ , where  $k$  is an integer key to be stored in the node, and  $l$  and  $r$  are bits (1/0) indicating whether the node has a left child and a right child, or not. The user specifies the triples in a level-by-level and left-to-right (in each level) fashion. One sample input is shown as below.

```
297 1 1
319 0 1
124 1 1
282 0 0
530 1 1
424 0 0
287 1 1
214 0 0
471 0 0
376 0 1
173 0 0
```

### 1.2 Construction of left-tilted tree from the binary tree

Once  $T_{bt}$  has been constructed then the next step is to construct the left-tilted tree or  $T_{ltt}$  which is defined as follows-

- A binary tree  $T_{bt}$  is called left-tilted tree or  $T_{ltt}$  if for any node  $v$  in  $T_{bt}$ , we have  $ht(L(v)) \geq ht(R(v))$ . Here  $L(v)$  and  $R(v)$  indicate left and right subtrees with root node  $v$  respectively.
- Thus, while constructing  $T_{ltt}$ , if a node  $v$  is encountered such that  $ht(L(v)) < ht(R(v))$  then it is required to swap the two child links.

The resultant left tilted tree for the sample binary tree is given below.

```
297 1 1
124 1 1
319 1 0
530 1 1
424 0 0
282 0 0
287 1 1
214 0 0
376 1 0
471 0 0
173 0 0
```

### 1.3 Printing the tree

Write a function `printtree()` to print a binary tree  $T_{bt}$  using preorder and inorder traversal manner in an output file `op.txt`.

For the input binary tree

Preorder: 297 319 282 124 530 287 471 376 173 214 424

Inorder: 319 282 297 471 287 376 173 530 214 124 424

For the left tilted binary tree

Preorder: 297 124 530 287 376 173 471 214 424 319 282

Inorder: 173 376 287 471 530 214 124 424 297 282 319

## 2 Submission Guidelines

- i Mention the strategy and any other assumptions that you have used in the `assign4README.txt` file. Include the running time complexity and extra space requirement complexity of your strategy in this file with necessary justification.
- ii 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. Avoid using global or static variables.
- iii We will evaluate the codes using `gcc` or `g++` commands at the terminal mode.
- iv After the due date and time (mentioned at top right with red font), the submission will remain open for 12 hours more. However, submission after due time will be treated as late submission and there will be 20% penalty for such late submission. As lab instructor or the TAs may not be available to fix the login/ networking problem at the last moment so upload the assignment well in advance to avoid any last minute glitches.
- v There will be penalty if you are found to take any unfair means during the lab hours and during the assignment submission process.
- vi Copying others' program and allowing others to copy your program will be equally penalized.