CS214: Data Structures 2017
Assignment (4) V1.0
[120 points]

Deadline & Submission

1. At least one team member should submit the compressed group solution as zip file containing the programs under acadox –> tasks (name your assignment file "A4_ID1_ID2.zip").
2. The deadline for submitting the electronic solution of this assignment is ***Sunday 30th April 2017 at 10:00 pm.***

About this assignment

1. This assignment will be solved in teams of 2.
2. The weight of the assignment is 120 points
3. All team members should understand all assignment problems.
4. All code must be standard ANSI C++.
5. Assume any missing details and search for information yourself.
6. Any cheating in any part of the assignment is the responsibility of the whole team and whole team will be punished.

Problems

Section#1 (50 points):

Create template binary search tree class with this name BSTFCI and create node class with name BSTNode (10 points)

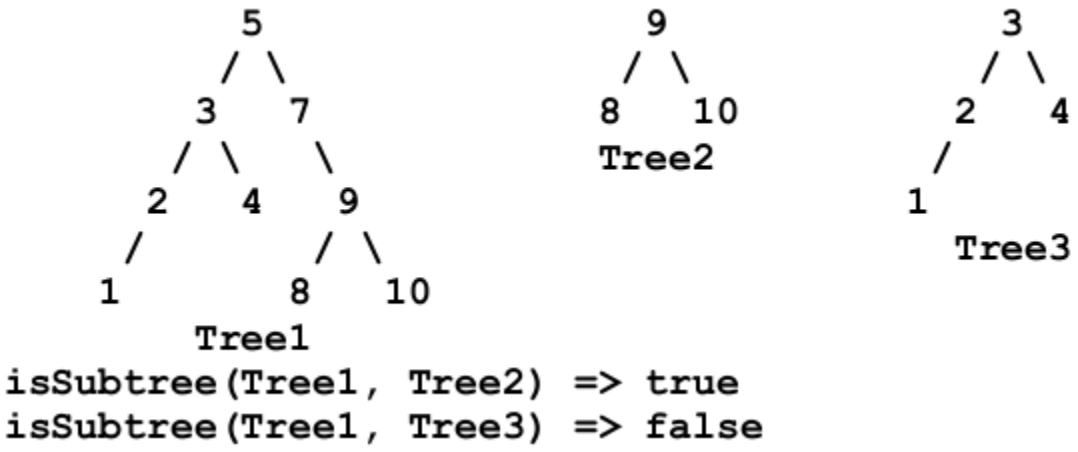
1. Add Checking Tree Balance (10 points)

A Balanced Binary Tree is a tree where the heights of the two child sub-trees of any node differ by at most one AND the left subtree is balanaced AND the right subtree is balanced.

Add method called 'isBalance' to BSTFCI this method will check in the BST is balance or not.

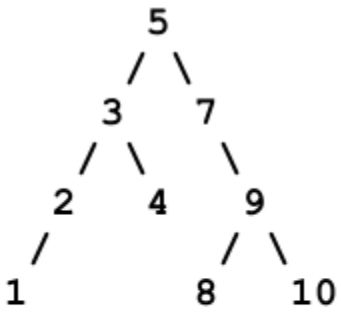
2. Tree Comparison (10 points)

Write a function that decides if a BSTFCI T2 is a sub-tree of another BSTFCI T1.
Prototype:
bool isSubTree(BSTFCI* t1, BSTFCI* t2);
Note: You may need to write another function that takes 2 BSTNodes and compares their sub-trees.



3. Print Range (10 points)

Add a recursive function named printRange in the class BSTFCI that stores integers and given a low key value and a high key value, it prints in sorted order all records whose key values fall between the two given keys.
Function printRange should visit as few nodes in the BST as possible.
You should NOT traverse ALL the tree inorder and print the ones in the range. This will not be considered a correct solution. You should do smart traversal to only traverse the related parts.




```
printRange(3, 6)    => [3,4,5]
printRange(8, 15)   => [8,9,10]
printRange(6,6)     => []
```

4. Tree Application (10 points)

- Write an index builder application that takes text consisting of lines and prints a list of the words of the text and the lines they appear on are printed next to them.
- The application works by building a binary search tree using BSTFCI and each node contains a word and a vector of that contains the list of lines where this word exists. For each new word, the program finds it and adds the line number to the vector. If word is not found, it is added to the tree. Then traverse the tree in-order to print the nodes.
- You need to remove punctuation marks like . and , from the text before processing it.
- For example, the text below produces the given index, Test Your code on the given text and 1 more examples.

I am for truth, no matter who tells it. I am for justice, no matter who it is for or against. Malcolm X

against	4	matter	4
am	1, 3	no	2, 4
for	1, 3, 4	or	4
I	1, 3	tells	2
is	4	truth	1
it	2, 4	who	2,4
justice	3	X	5
Malcolm	5		

Section#2 (35 points)

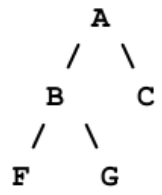
1. Expression Tree Evaluation (10 points)

- Write an algorithm that accepts an arithmetic expression written in prefix notation and builds an expression tree for it. And then traverse to evaluate the expression. The evaluation should start after the whole expression has been entered.
- Implement your algorithm as a function.
- Write five test cases for your application and run them.

```
expression: + 3 * 4 / 8 2
tree:
          +
         / \
        3   *
           / \
          4   /
             / \
            8   2
Evaluation: 19
```

2. Tree Traversal (15 points)

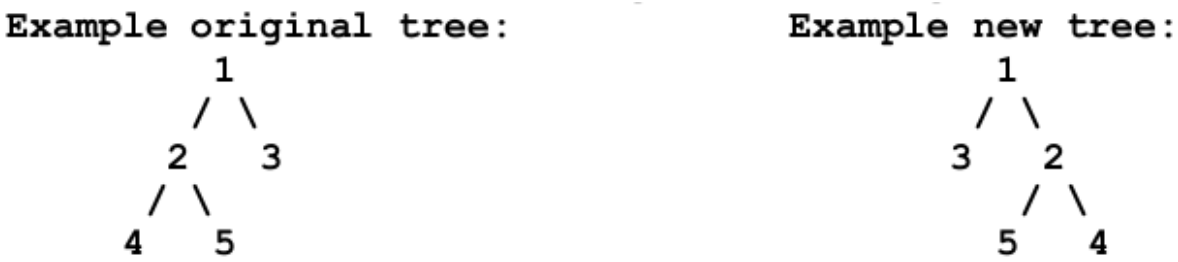
- Assume a binary tree of non-repeated characters. Write a function printPostOrder that takes two strings representing the preorder traversal and the in-order traversal of the tree. Then the function prints the post-order traversal. The function prototype is:
  void printPostOrder (string preorder, string inorder);
- Write a program that implements five test cases to test your function.
- A sample function call and the corresponding output is shown below:
  printPostOrder ("ABFGC", "FBGAC")
  => FGBCA
  // Tree is



3. Tree Flipping (10 points)

Assume a binary tree. Write a flip method that takes the node which the mirror image of the tree will start from if no parameter send to the function the default value will be the root node.

void flip(Node* node = root)



Section#3: (35 point)

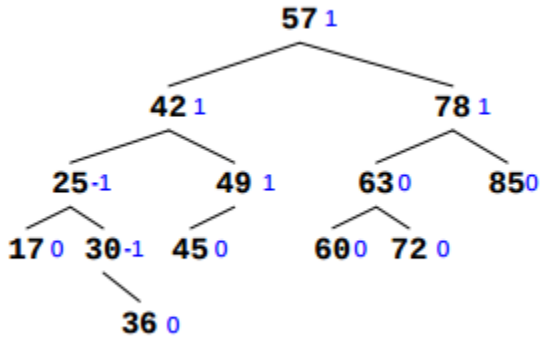
Problem#1 Implement BST AVL tree (23 points)

1.Insert Element into the tree (6 points)
2.Delete Element from tree (14 points)
3.Display Balanced AVL Tree: (5 points)
- InOrder traversal
- PreOrder traversal
- PostOrder traversal

Problem#2 (2 points)

Explain on the BST AVL what will happen when you add element with value 61
Note: for solve this problem you should view what happen in two steps and show the changes in balance factors
Step1: just add the node to the tree (0.5 point)
Step2: make the tree balance(1.5 points)

Balance factors shown next to each node in blue



Problem#3 (8 points)

Apply the following operations on the BST
Note: The questions are independent on each other for solving any question you should do two steps
Step1: redraw the tree after applying the operation (50%)
Step2: explanation for your step (50%)

1) Delete the node with value 60 (2 points)
2) Delete the node with value 78 (3 points)
3) Delete the node with value 42 (3 points)