Project 1

CMPE 250, Data Structures and Algorithms, Spring 2011

Instructor: A. T. Cemgil
TA’s: Cem Keskin, Cihat İmamoğlu

Due: April 13th, 23:59

A4.1 Problem Definition

In this programming assignment, you are to find the post-order traversal of a binary tree given its in-order and pre-order traversals. Below, the definitions of these traversals are given.

Pre-order traversal: visit the root first; and then traverse the left subtree; and then traverse the right subtree.

In-order traversal: traverse the left subtree; and then visit the root; and then traverse the right subtree.

Post-order traversal: traverse the left subtree; and then traverse the right subtree; and then visit the root.

Assignment

• Write an ANSI C++ program that does what’s described above. Come up with an algorithm and implement it. You must use appropriate data structures and OOP aspects you learned in this lecture.

Input Format (tree.in)

• The first line of the input will be a positive integer T (<= 2000), denoting the number of test cases.
• Each of the next T lines of input will have the following: a number N (1 <= N <= 52), denoting the number of nodes in the tree; a space; the string S1, denoting the pre-order traversal of the tree; another space; the string S2, denoting the in-order traversal of the tree. The nodes of the tree are labeled with different characters in the range a-z and A-Z.

Output Format (tree.out)

• There will be T lines in your output file. Each line will have the post order-traversal of its corresponding tree.

Example Input

3
3 xYz Yxz
3 abc cba
6 ABCDEF CBAEDF
Example Output

Yzx
cba
CBEFDA

Important Points

- It is very important that you follow exactly the given input/output format.
- The input will always be in the valid format described above, i.e. you do not have to modify your program for a possible malformatted input.

A4.2 Bonuses

- Devise and code an algorithm for the problem using a stack, i.e. not using any recursion.
- Prove that given pre-order and in-order traversals represent a unique, unambiguous binary tree.
- Prove/disprove: given a pre-order and post-order traversals, we can build an unambiguous tree.

Submission & Grading

- What to submit
  - Your commented C++ codes, exemplary input and output files as described above. If you attempt any part of the Bonus section, submit a short report describing what you have done, the proofs etc..

- How and when to submit
  - Due date is April 13th, 23:59. Start early so that you can ask questions if you have any problems. Compress the files and name them: 'Name Surname-Cmpe250-Project1.zip' (or rar) and send them to keskinc@cmpe.boun.edu.tr.

- Grading
  - Your program will be graded based on the quality of your source code and correctness of your output. Correctness of your output will be tested automatically so make sure you stick with the format described above.
  - Make sure you document your code with necessary inline comments, and use meaningful variable names. Do not overcomment, or make your variable names unnecessarily long.
  - If you solve any of the bonuses, create a short document that describes your algorithms/proofs.