CS 014: Introduction to Data Structures and Algorithms
Fall 2017
Lab 3

Objectives:

In this lab, your task is simple. You need to implement preorder traversal and inorder traversal for a binary tree *without* recursion.

Deliverables:

- (10%) You have to attend the lab on Monday.
- (50%) The final source code.
- (15%) A brief report that describes the data structures and algorithms that you used in your program. At the top of your report, please indicate your name, student ID, and your TA's name.
- (5%) You must adhere to the following submission format. You need to submit a single file on iLearn named "CS014_lab3_<UCR Net ID>.zip" where <UCR Net ID> is the student's UCR Net ID. The file should contain a PDF file for the report and the source code files in .cc, .hh, or whatever extension for source files you used.
- (20%) There will be a relevant question during the lab on Monday 10/23 that you will be asked to do during the lab.

Groups:

- This lab should be done individually.

Due date:

- The deliverables are due on Tuesday 10/24 by 11:59 PM Pacific Time. However, you are highly encouraged to deliver it during the lab on Monday 10/23 to save your time.

Important Note:

- You will need to carry out this lab at home to be able to perform the task that will be given to you during the lab on Monday 10/23. The official due date is on Tuesday just in case you have questions for your TA during the lab.

Problem definition:

Two of the fundamental traversals for binary trees are perorder traversal and inorder traversal. A preorder traversal can be defined recursively as the concatenation of:

- The label (value) of the root.
- The preorder traversal of the left subtree.
- The preorder traversal of the right subtree.

The preorder traversal of the empty tree is the empty set.

Similarly, the inorder traversal can be defined recursively as the concatenation of:

- The inorder traversal of the left subtree.
- The label (value) of the root.
- The inorder traversal of the right subtree.

The inorder traversal of the empty tree is the empty set.

For example, the following tree has an inorder traversal of {D, B, E, A, C, F} and a preorder traversal of {A, B, D, E, C, F}.

```
    A
   / \
  B   C
 / \   \
D   E   F
```

As in the previous lab, the initial code runs some tests on tree examples and tests their preorder and inorder traversals. In addition, the code already has a recursive implementation for both traversals that you can use as a reference. Your task is to modify the implementations of the two functions to remove any recursive calls. *(Hint: You will need to use a stack in place of the recursion.)*

Detailed requirements for each function:

- The input is a pointer to the root of a binary tree.
- You can find the definition of the binary tree node in the file "helper.hh".
- The output is a vector of values that contains either the preorder or inorder traversal.
- You are allowed to use the following data structures from the STL library:
    - stack
    - vector
    - pair

Steps:

1. Create a workspace in Cloud9 and initialize it with the code that is attached to this lab. You can do this by pasting the URL 'https://github.com/aseldawy/CS014-Lab3' in the field 'Clone from Git or Mercurial URL.' Instead, you can download the source code from iLearn and upload it to an empty workspace in Cloud9.
2. To compile the code, type "`make`" in the shell. In Cloud9, you can access the shell which is usually at the bottom of the screen.
3. To run the code, type "`./traversals`" in the shell.
4. The code uses Google Test library to test a few basic cases for preorder and inorder traversals. Initially, all the tests pass. Your job is to rewrite the two traversal functions so that the tests still pass.
5. At delivery, we might use additional test cases other than the ones listed in the source code. Try to make your source code robust by making additional tests.