Middle East Technical University | Department of Computer Engineering

CENG 242

Programming Languages

Spring '2013-2014

Programming Assignment 1

Due date: 16 March 2014, Sunday, 23:55

1 Objectives

This assignment aims you to get familiar with basic functional programmimg concepts by using Haskell.

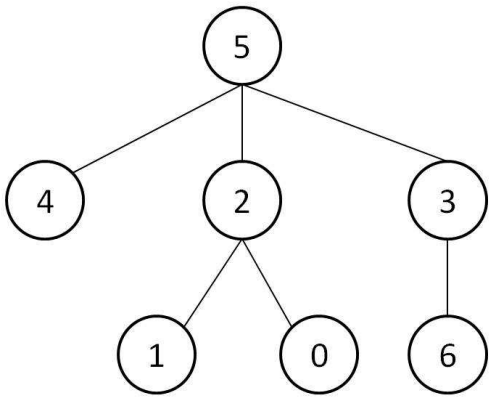
2 Specifications

In this homework, you are going to be given preorder and postorder traversals of an n-ary tree as input. Then you are expected to construct and show the tree corresponding these traversals.

You are going to use the below data type to represent an n-ary tree:

```
data Tree = Empty | Node Integer [Tree] deriving Show
```

That is, nodes of the n-ary tree will include Integer numbers only. "Node Integer [Tree]" represents a node whose value is specifed in Integer part, and subtrees (children) of that node is specified in [Tree] part as a list of Trees. "Empty" means the end of that branch. For example, Node 5 [Node 4 [Empty], Node 2 [Node 1 [Empty], Node 0 [Empty]], Node 3 [Node 6 [Empty]]] represents the tree below:



Here, the node with the value 4, the node with the value 1, the node with the value 0 and the node with the value 6 are leaves of the tree, as it is seen.

The nodes of the n-ary trees in the homework will be Integers with different values.

You are going to write a function constructTree in the form of:

`constructTree :: [Integer] -> [Integer] -> Tree`

That is, constructTree function will take 2 arguments where the first argument is a list of integers representing the preorder traversal of the tree and the second argument is also a list of integers representing the postorder traversal of the tree. In the end, the function will return the corresponding tree in the data type "Tree" which was defined above.

3 Examples

1. **Input:**
   constructTree [60, 20, 10, 40, 30 ,50, 70] [10, 30, 50, 40, 20, 70, 60]
   Output:
   Node 60 [Node 20 [Node 10 [Empty],Node 40 [Node 30 [Empty],Node 50 [Empty]]],Node 70 [Empty]]

2. **Input:**
   constructTree [1, 2, 5, 3, 6, 10, 11, 7, 4, 8, 9] [5, 2, 11, 10, 6, 7, 3, 8, 9, 4, 1]
   Output:
   Node 1 [Node 2 [Node 5 [Empty]],Node 3 [Node 6 [Node 10 [Node 11 [Empty]]],Node 7 [Empty]],Node 4 [Node 8 [Empty],Node 9 [Empty]]]

3. **Input:**
   constructTree [5, 2, 1, 8, 7, 6, 12, 10, 9, 13, 15] [1, 2, 6, 7, 9, 10, 15, 13, 12, 8, 5]
   Output:
   Node 5 [Node 2 [Node 1 [Empty]],Node 8 [Node 7 [Node 6 [Empty]],Node 12 [Node 10 [Node 9 [Empty]],Node 13 [Node 15 [Empty]]]]]

4 Regulations

1. **Programming Language:** Haskell
2. **Late Submission:** You have 3 days out of 10 (total lates for all homeworks) for the late submission.
3. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating will receive zero from all homework (hence get NA from this course).
4. **Newsgroup:** You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.
5. **Evaluation:** Your program will be evaluated automatically using "black-box" technique so make sure to obey the specifications. Your codes will also be visually investigated and code analysis tool will be used to catch similarities between submissions. For that reason, never share your code with others.

5 Submission

Submission will be done via COW. Create a tar document named `hw1.tar.gz` which contains a single file named `hw1.hs` that contains your code.
Note: The submitted archive should not contain any directories! Failing to obey this specification will cause deduction in your grades.