Given a tree | 3 point(s)

Overview

In computer science, trees are a very interesting data structure with many different algorithmic applications. This problem focuses specifically on [Binary Trees](http://en.wikipedia.org/wiki/Binary_tree).

A binary tree is a tree structure in which each node has at most two child nodes (often referred to as a left and right).[[1](http://en.wikipedia.org/wiki/Binary_tree)]

There are several different ways to traverse an existing tree, but in this problem we involve 3 specific traversals.

* [In-order](http://en.wikipedia.org/wiki/In-order_traversal)
* [Post-order](http://en.wikipedia.org/wiki/Post-order_traversal)
* [Breadth-first](http://en.wikipedia.org/wiki/Breadth-first_search)

In this problem, you will be provided a binary tree as input. The tree will be represented by in-order and post-order traversals.

The task is to build up a binary tree solely from the given traversals and output a breadth-first traversal.

The input tree can be invalid, meaning that it does not represent a binary tree.

You have to make sure to handle invalid input and output InvalidInput if you determine the input to be invalid.

An example of invalid input could be something like this 1-2-3-4-5-6 or this is not a valid binary tree input.

See the input and output for more examples of valid and invalid inputs.

Clarifications

* Tree node values are unique, so don’t worry about ambiguity caused by duplicate values
* If the tree or input is invalid, the result should be the string InvalidInput
* The input trees are not balanced
* The input trees have no node value rules
* These are binary trees, so remember, at most two children per node

Input description/format

A string consisting of two lines.

Each line is a comma separated list of numbers, representing the in-order and post-order traversals.

The numbers themselves are the node values.

Output description/format

A string representing the breadth-first-order traversal of the tree, where node values are comma separated to represent the tree nodes.

Example input

4,2,7,5,8,1,3,9,6,11,10

4,7,8,5,2,9,11,10,6,3,1

Example output

1,2,3,4,5,6,7,8,9,10,11