The Great Tree-List Recursion Problem.

Asked by Varun Bhatia.

Question:

Write a recursive function treeToList(Node root) that takes an ordered binary tree and rearranges the internal pointers to make a circular doubly linked list out of the tree nodes. The"previous" pointers should be stored in the "small" field and the "next" pointers should be stored in the "large" field. The list should be arranged so that the nodes are in increasing order. Return the head pointer to the new list.

This is very well explained and implemented at http://cslibrary.stanford.edu/109/TreeListRecursion.html

102 Comments Category: Linked Lists Trees

Related Posts:

- Merge two sorted linked lists such that merged list is in reverse order
- Compare two strings represented as linked lists
- Rearrange a given linked list in-place.
- Sort a linked list that is sorted alternating ascending and descending orders?

- Select a Random Node from a Singly Linked List
- Merge Sort for Doubly Linked List
- Point to next higher value node in a linked list with an arbitrary pointer
- Swap nodes in a linked list without swapping data

(Login to Rate and Mark)

4.5 Average Difficulty : **4.5/5.0**
Based on **2** vote(s)

Add to TODO List

Mark as DONE

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.