Trees transformation

Team ID: 23
Team Member: B03202010 B03202020 B03202054

1. ## Topic

   Left-child right-sibling representation, tree traversal

2. ## Description

   We learned in class that a general tree can be transformed into a binary tree by Left-child right-sibling method. Given a transformed binary tree, try to get three orders (preorder, postorder, level order) of the original general tree.

   The given binary tree is based on vector, with unused nodes being -1.

3. ## Input and output format

   Input:
   The first line shows an integer H, meaning the height of the tree.
   For a tree only have root, the height is 1.
   The second line shows 2^H - 1 integers, being the vector based BT.

   For first testdata file, 7 <= H <= 12 with almost 10000 trees.
   For second testdata file, 15 <= H <= 20 with almost 50 trees.
   For third testdata file, H = 24 with only 1 tree.

   Output:
   Preorder: “preorder of the GT transformed from input BT”
   Postorder: “Postorder of the GT transformed from input BT”
   Level order: “Level order of the GT transformed from input BT”

   For the input H = 0, meaning the end of the testdata, don’t output anything.

   One line for every kind of the output and another blank line between every tree.
   Don’t output “, and be careful for the space.

   Be careful for TLE.

4. ## Sample input and output

   Input:
   5
   1 2 -1 5 3 -1 -1 -1 6 7 4 -1 -1 -1 -1 -1 -1 -1 -1 11 8 10 -1 -1 -1 -1 -1 -1 -1 -1 -1
   4
   9 4 -1 -1 8 -1 -1 -1 -1 -1 7 -1 -1 -1 -1
   0

   Output:
   preorder: 1 2 5 6 3 7 11 8 4 10
   postorder: 5 6 2 11 7 8 3 10 4 1

level order: 1 2 3 4 5 6 7 8 10 11

preorder: 9 4 8 7
postorder: 4 8 7 9
level order: 9 4 8 7

5. Time and memory limit

7 sec, 100MB