Trees 3

20 March 2020 10:13

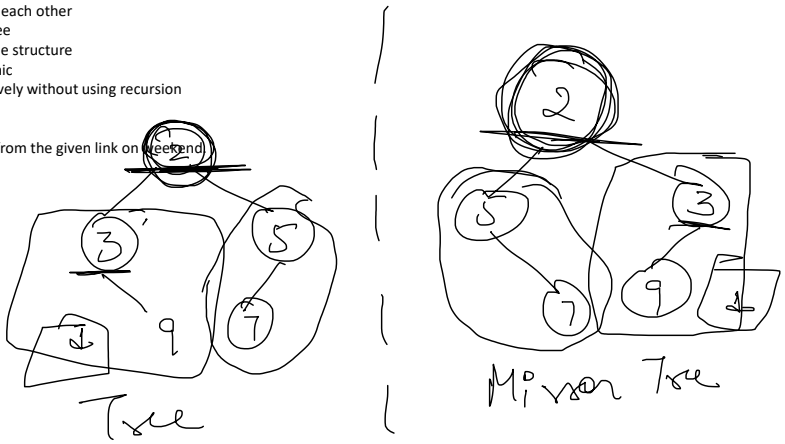
Content:

20-March-2020

1. Check if two trees are mirror to each other
2. Convert a tree into its mirror tree
3. Check if two trees have the same structure
4. Check if two trees are IsoMorphic
5. Get the height of a tree itereatively without using recursion

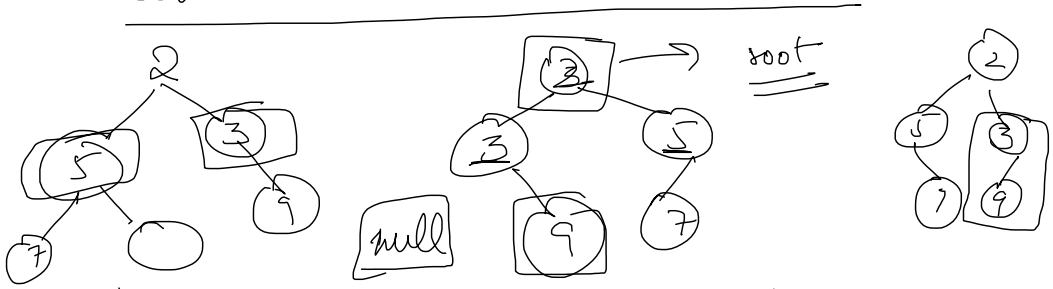
H.W

1. Find the level of a given node.
2. Complete at least first 20 ques from the given link on (weekend)



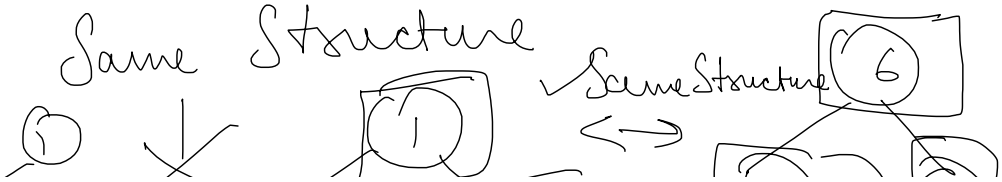

```
boolean isMirror (Node root1, Node root2)
{   if(root1==null && root2==null) return true;
    if(root1==null || root2==null) return false;
return root1.data == root2.data &&
    isMirror(root1.left, root2.right) &&
    isMirror(root1.right, root2.left);
```

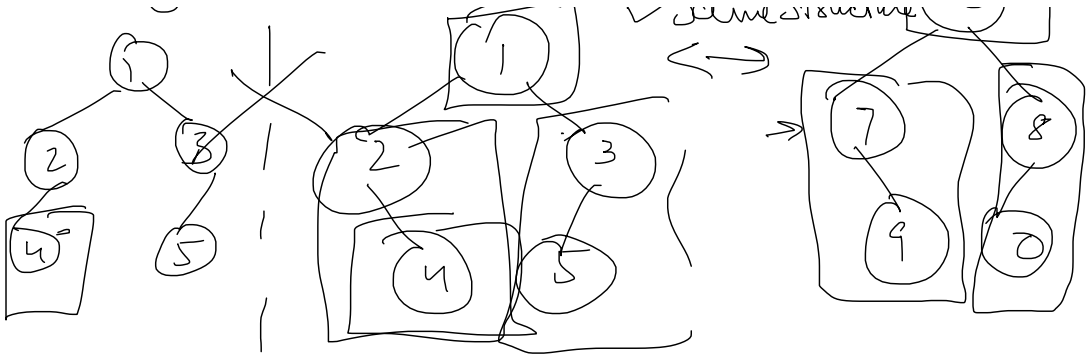
Convert a tree to its Mirror



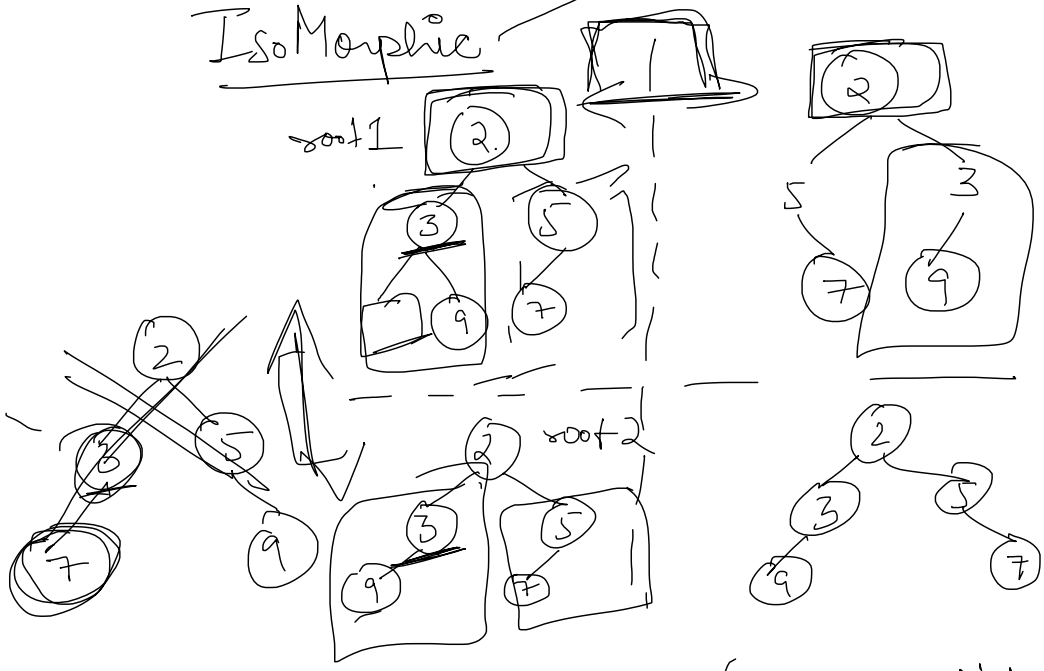

```
Node toMirror(Node root)
{   if(root==null){return null;}
    Node temp = root.left;
    root.left = root.right;
    root.right = temp;
    toMirror(root.left);
    toMirror(root.right);
    return root;
}
```

Same Structure





IsoMorphic



For each Node children are either swapped or in the same order

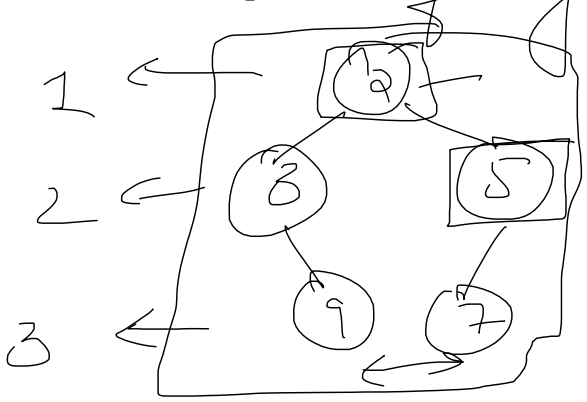
```
boolean isIsoMorphic (Node root1, Node root2)
{ if (root1 == null && root2 == null) return true;
  if (root1 == null || root2 == null) return false;
  if (root1.data != root2.data)
    { return false; }
  return ( isIsoMorphic(root1.left, root2.left) &&
           isIsoMorphic(root1.right, root2.right) ) ||
         ( isIsoMorphic(root1.left, root2.right) &&
           isIsoMorphic(root1.right, root2.left) )
```

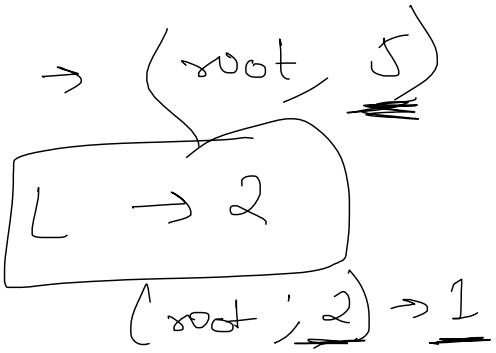
Get the height of a tree iteratively without using recursion

without using recursion

H.W

Get the level of a given Node





(root, 9) (root, 7)

3

1 → Pass

2

3 5