

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1	Solution 2
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 import java.util.ArrayDeque; 4 5 class Program { 6 // O(n) time O(n) space 7 public static void invertBinaryTree(BinaryTree tree) { 8 ArrayDeque<BinaryTree> queue = new ArrayDeque<BinaryTree>(); 9 queue.addLast(tree); 10 while (queue.size() > 0) { 11 BinaryTree current = queue.pollFirst(); 12 if (current == null) { 13 continue; 14 } 15 swapLeftAndRight(current); 16 if (current.left != null) { 17 queue.addLast(current.left); 18 } 19 if (current.right != null) { 20 queue.addLast(current.right); 21 } 22 } 23 } 24 25 private static void swapLeftAndRight(BinaryTree tree) { 26 BinaryTree left = tree.left; 27 tree.left = tree.right; 28 tree.right = left; 29 } 30 31 static class BinaryTree { 32 public int value; 33 public BinaryTree left; 34 public BinaryTree right; 35 36 public BinaryTree(int value) { 37 this.value = value; 38 } 39 } 40 } 41</pre>	<pre>1 class Program { 2 public static void invertBinaryTree(BinaryTree tree) { 3 // Write your code here. 4 } 5 6 static class BinaryTree { 7 public int value; 8 public BinaryTree left; 9 public BinaryTree right; 10 11 public BinaryTree(int value) { 12 this.value = value; 13 } 14 } 15 } 16</pre>

Run or submit code when you're ready.