

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1	Solution 2	Solution 3
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 using System.Collections.Generic; 4 5 public class Program { 6     // O(n) time   O(n) space 7     public static void InvertBinaryTree(BinaryTree tree) { 8         List&lt;BinaryTree&gt; queue = new List&lt;BinaryTree&gt;(); 9         queue.Add(tree); 10        var index = 0; 11        while (index &lt; queue.Count) { 12            BinaryTree current = queue[index]; 13            index += 1; 14            if (current == null) { 15                continue; 16            } 17            swapLeftAndRight(current); 18            if (current.left != null) { 19                queue.Add(current.left); 20            } 21            if (current.right != null) { 22                queue.Add(current.right); 23            } 24        } 25    } 26 27    private static void swapLeftAndRight(BinaryTree tree) { 28        BinaryTree left = tree.left; 29        tree.left = tree.right; 30        tree.right = left; 31    } 32 33    public class BinaryTree { 34        public int value; 35        public BinaryTree left; 36        public BinaryTree right; 37 38        public BinaryTree(int value) { 39            this.value = value; 40        } 41    } 42 } 43</pre>		<pre>1 2 public class Program { 3     public static void InvertBinaryTree(BinaryTree tree) { 4         // Write your code here. 5     } 6 7     public class BinaryTree { 8         public int value; 9         public BinaryTree left; 10        public BinaryTree right; 11 12        public BinaryTree(int value) { 13            this.value = value; 14        } 15    } 16 } 17</pre>

**Run or submit code when you're ready.**

---