

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1	Solution 2
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 #include <vector> 4 #include <deque> 5 using namespace std; 6 7 class BinaryTree { 8 public: 9 int value; 10 BinaryTree *left; 11 BinaryTree *right; 12 13 BinaryTree(int value); 14 void insert(vector<int> values, int i = 0); 15 void invertedInsert(vector<int> values, int i = 0); 16 }; 17 18 void swapLeftAndRight(BinaryTree *tree); 19 20 // O(n) time O(n) space 21 void invertBinaryTree(BinaryTree *tree) { 22 deque<BinaryTree *> queue; 23 queue.push_back(tree); 24 while (queue.size() > 0) { 25 BinaryTree *current = queue.front(); 26 queue.pop_front(); 27 if (current == NULL) { 28 continue; 29 } 30 swapLeftAndRight(current); 31 queue.push_back(current->left); 32 queue.push_back(current->right); 33 } 34 } 35 36 void swapLeftAndRight(BinaryTree *tree) { 37 BinaryTree *left = tree->left; 38 tree->left = tree->right; 39 tree->right = left; 40 } 41</pre>	<pre>1 #include <vector> 2 using namespace std; 3 4 class BinaryTree { 5 public: 6 int value; 7 BinaryTree *left; 8 BinaryTree *right; 9 10 BinaryTree(int value); 11 void insert(vector<int> values, int i = 0); 12 void invertedInsert(vector<int> values, int i = 0); 13 }; 14 15 void invertBinaryTree(BinaryTree *tree) { 16 // Write your code here. 17 } 18</pre>

Run or submit code when you're ready.