

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 using namespace std;
5
6 class BinaryTree {
7 public:
8     int value;
9     BinaryTree *left;
10    BinaryTree *right;
11
12    BinaryTree(int value);
13    void insert(vector<int> values, int i = 0);
14    void invertedInsert(vector<int> values, int i = 0);
15 };
16
17 void swapLeftAndRight(BinaryTree *tree);
18
19 // O(n) time | O(d) space
20 void invertBinaryTree(BinaryTree *tree) {
21     if (tree == NULL) {
22         return;
23     }
24     swapLeftAndRight(tree);
25     invertBinaryTree(tree->left);
26     invertBinaryTree(tree->right);
27 }
28
29 void swapLeftAndRight(BinaryTree *tree) {
30     BinaryTree *left = tree->left;
31     tree->left = tree->right;
32     tree->right = left;
33 }
34
```

Solution 1

Solution 2

Solution 3

```
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
```

Custom Output

Raw Output

Submit Code

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