Our Solution(s) Run Code

Solution 1 Solution 2

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

#include <vector>
#include <deque>
using namespace std;

class BinaryTree {
public:
  int value;
  BinaryTree *left;
  BinaryTree *right;

  BinaryTree(int value);
  void insert(vector<int> values, int i = 0);
  void invertedInsert(vector<int> values, int i = 0);
};

void swapLeftAndRight(BinaryTree *tree);

// O(n) time | O(n) space
void invertBinaryTree(BinaryTree *tree) {
  deque<BinaryTree *> queue;
  queue.push_back(tree);
  while (queue.size() > 0) {
    BinaryTree *current = queue.front();
    queue.pop_front();
    if (current == NULL) {
      continue;
    }
    swapLeftAndRight(current);
    queue.push_back(current->left);
    queue.push_back(current->right);
  }
}

void swapLeftAndRight(BinaryTree *tree) {
  BinaryTree *left = tree->left;
  tree->left = tree->right;
  tree->right = left;
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
#include <vector>
using namespace std;

class BinaryTree {
public:
  int value;
  BinaryTree *left;
  BinaryTree *right;

  BinaryTree(int value);
  void insert(vector<int> values, int i = 0);
  void invertedInsert(vector<int> values, int i = 0);
};

void invertBinaryTree(BinaryTree *tree) {
  // Write your code here.
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.