AlgoExpert Quad Layout C++ 12px Sublime Monokai 00:00:0

Our Solution(s) Run Code

Solution 1

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

#include <vector>
using namespace std;

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

  BST(int val);
};

// O(n) time | O(n) space
vector<int> inOrderTraverse(BST *tree, vector<int> array) {
  if (tree->left != NULL) {
    array = inOrderTraverse(tree->left, array);
  }
  array.push_back(tree->value);
  if (tree->right != NULL) {
    array = inOrderTraverse(tree->right, array);
  }
  return array;
}

// O(n) time | O(n) space
vector<int> preOrderTraverse(BST *tree, vector<int> array) {
  array.push_back(tree->value);
  if (tree->left != NULL) {
    array = preOrderTraverse(tree->left, array);
  }
  if (tree->right != NULL) {
    array = preOrderTraverse(tree->right, array);
  }
  return array;
}

// O(n) time | O(n) space
vector<int> postOrderTraverse(BST *tree, vector<int> array) {
  if (tree->left != NULL) {
    array = postOrderTraverse(tree->left, array);
  }
  if (tree->right != NULL) {
    array = postOrderTraverse(tree->right, array);
  }
  array.push_back(tree->value);
  return array;
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

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

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

  BST(int val);
};

vector<int> inOrderTraverse(BST *tree, vector<int> array) {
  // Write your code here.
  return {};
}

vector<int> preOrderTraverse(BST *tree, vector<int> array) {
  // Write your code here.
  return {};
}

vector<int> postOrderTraverse(BST *tree, vector<int> array) {
  // Write your code here.
  return {};
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.