Prompt   Scratchpad   Our Solution(s)   Video Explanation   Run Code

Your Solutions   Run Code

Solution 1

Solution 1   Solution 2   Solution 3

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

using System.Collections.Generic;

public class Program {
  // O(n) time | O(n) space
  public static List<int> InOrderTraverse(BST tree, List<int> array) {
    if (tree.left != null) {
      InOrderTraverse(tree.left, array);
    }
    array.Add(tree.value);
    if (tree.right != null) {
      InOrderTraverse(tree.right, array);
    }
    return array;
  }

  // O(n) time | O(n) space
  public static List<int> PreOrderTraverse(BST tree, List<int> array) {
    array.Add(tree.value);
    if (tree.left != null) {
      PreOrderTraverse(tree.left, array);
    }
    if (tree.right != null) {
      PreOrderTraverse(tree.right, array);
    }
    return array;
  }

  // O(n) time | O(n) space
  public static List<int> PostOrderTraverse(BST tree, List<int> array) {
    if (tree.left != null) {
      PostOrderTraverse(tree.left, array);
    }
    if (tree.right != null) {
      PostOrderTraverse(tree.right, array);
    }
    array.Add(tree.value);
    return array;
  }

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

    public BST(int value) {
      this.value = value;
    }
  }
}
```

```csharp
using System.Collections.Generic;

public class Program {
  public static List<int> InOrderTraverse(BST tree, List<int> array) {
    // Write your code here.
    return null;
  }

  public static List<int> PreOrderTraverse(BST tree, List<int> array) {
    // Write your code here.
    return null;
  }

  public static List<int> PostOrderTraverse(BST tree, List<int> array) {
    // Write your code here.
    return null;
  }

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

    public BST(int value) {
      this.value = value;
    }
  }
}
```

Custom Output   Raw Output   Submit Code

Run or submit code when you're ready.