

Our Solution(s)Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 type BinaryTree struct {
6     Value int
7
8     Left *BinaryTree
9     Right *BinaryTree
10 }
11
12 // O(n) time | O(d) space
13 func (tree *BinaryTree) InvertBinaryTree() {
14     tree.Left, tree.Right = tree.Right, tree.Left
15     if tree.Left != nil {
16         tree.Left.InvertBinaryTree()
17     }
18     if tree.Right != nil {
19         tree.Right.InvertBinaryTree()
20     }
21 }
22
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 package main
2
3 type BinaryTree struct {
4     Value int
5
6     Left *BinaryTree
7     Right *BinaryTree
8 }
9
10 func (tree *BinaryTree) InvertBinaryTree() {
11     // Write your code here.
12 }
13
```

```

10 return 0
11
12 "Prints out the value of the variable 'speed'"
13
14
15 def calculate_mph(speed, value ... 200): "converts"
16     mph = (speed * 1.60934)
17     for i, value in enumerate(speed):
18         print(speed[i])
19     return mph
20
21
22 def main():
23     pass
24
25
26 def main(): "converts speed to mph"
27     speed = 100
28     for i, value in enumerate(speed):
29         mph = speed[i]
30         print(mph)
31         return mph
32
33
34

```

Run or submit code when you're ready.