

Prompt	Scratchpad	Our Solution(s)	Video Explanation	Run Code
--------	------------	-----------------	-------------------	----------

Solution 1	Solution 2	Solution 3
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 package main 4 5 // O(n) time O(n) space - where n is the length of the array 6 ▾ func MinHeightBST(array []int) *BST { 7 return constructMinHeightBst(array, nil, 0, len(array)-1) 8 } 9 10 ▾ func constructMinHeightBst(array []int, bst *BST, startIdx, endIdx int) *BST { 11 ▾ if endIdx < startIdx { 12 return nil 13 } 14 midIdx := (startIdx + endIdx) / 2 15 newBstNode := &BST{Value: array[midIdx]} 16 ▾ if bst == nil { 17 bst = newBstNode 18 } else { 19 ▾ if array[midIdx] < bst.Value { 20 bst.Left = newBstNode 21 bst = bst.Left 22 ▾ } else { 23 bst.Right = newBstNode 24 bst = bst.Right 25 } 26 } 27 constructMinHeightBst(array, bst, startIdx, midIdx-1) 28 constructMinHeightBst(array, bst, midIdx+1, endIdx) 29 return bst 30 } 31 32 ▾ type BST struct { 33 Value int 34 35 Left *BST 36 Right *BST 37 } 38 39 // We don't use this method for this solution. 40 ▾ func (tree *BST) Insert(value int) *BST { 41 ▾ if value < tree.Value { 42 ▾ if tree.Left == nil { 43 tree.Left = &BST{Value: value} 44 ▾ } else { 45 tree.Left.Insert(value) 46 } 47 } else { 48 ▾ if tree.Right == nil { 49 tree.Right = &BST{Value: value} 50 ▾ } else { 51 tree.Right.Insert(value) 52 } 53 } 54 return tree 55 } 56</pre>		

