

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 // Average case: when the created BST is balanced 6 // O(nlog(n)) time   O(n) space - where n is the length of the array 7 // --- 8 // Worst case: when the the created BST is like a linked list 9 // O(n^2) time   O(n) space 10 func RightSmallerThan(array []int) []int { 11     if len(array) == 0 { 12         return []int{} 13     } 14 15     lastIdx := len(array) - 1 16     bst := NewSpecialBST(array[lastIdx], lastIdx, 0) 17     for i := lastIdx - 1; i &gt;= 0; i-- { 18         bst.Insert(array[i], i) 19     } 20 21     rightSmallerCounts := make([]int, 0, len(array)) 22     for _, i := range array { 23         rightSmallerCounts = append(rightSmallerCounts, i) 24     } 25     getRightSmallerCounts(bst, rightSmallerCounts) 26     return rightSmallerCounts 27 } 28 29 func getRightSmallerCounts(bst *SpecialBST, rightSmallerCounts []int) { 30     if bst == nil { 31         return 32     } 33     rightSmallerCounts[bst.Idx] = bst.NumSmallerAtInsertTime 34     getRightSmallerCounts(bst.Left, rightSmallerCounts) 35     getRightSmallerCounts(bst.Right, rightSmallerCounts) 36 } 37 38 type SpecialBST struct { 39     Value      int 40     Idx        int 41     NumSmallerAtInsertTime int 42     LeftSubtreeSize  int 43 44     Left *SpecialBST 45     Right *SpecialBST 46 } 47 48 func NewSpecialBST(value int, idx int, numSmallerAtInsertTime int) *SpecialBST { 49     return &amp;SpecialBST{ 50         Value:      value, 51         Idx:        idx, 52         NumSmallerAtInsertTime: numSmallerAtInsertTime, 53         LeftSubtreeSize: 0, 54         Left: nil, 55         Right: nil, 56     } 57 } 58 59 func (bst *SpecialBST) Insert(value, idx int) { 60     bst.insert(value, idx, 0) 61 } 62 63 func (bst *SpecialBST) insert(value, idx, numSmallerAtInsertTime int) { 64     if value &lt; bst.Value { 65         bst.LeftSubtreeSize += 1 66         if bst.Left == nil { 67             bst.Left = NewSpecialBST(value, idx, numSmallerAtInsertTime) 68         } else { 69             bst.Left.insert(value, idx, numSmallerAtInsertTime) 70         } 71         return 72     } 73     numSmallerAtInsertTime += bst.LeftSubtreeSize 74     if value &gt; bst.Value { 75         numSmallerAtInsertTime += 1 76     } 77 78     if bst.Right == nil { 79         bst.Right = NewSpecialBST(value, idx, numSmallerAtInsertTime) 80     } else { 81         bst.Right.insert(value, idx, numSmallerAtInsertTime) 82     } 83 }</pre>		

