

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

Solution 1	Solution 2	Solution 3
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 public class Program { 4     // O(d) time   O(1) space - where d is the depth (height) of the ancestral tree 5     public static AncestralTree GetYoungestCommonAncestor( 6         AncestralTree topAncestor, 7         AncestralTree descendantOne, 8         AncestralTree descendantTwo 9     ) { 10         int depthOne = getDescendantDepth(descendantOne, topAncestor); 11         int depthTwo = getDescendantDepth(descendantTwo, topAncestor); 12         if (depthOne &gt; depthTwo) { 13             return backtrackAncestralTree(descendantOne, descendantTwo, 14                 depthOne - depthTwo); 15         } else { 16             return backtrackAncestralTree(descendantTwo, descendantOne, 17                 depthTwo - depthOne); 18         } 19     } 20 21     public static int getDescendantDepth(AncestralTree descendant, AncestralTree to 22         int depth = 0; 23         while (descendant != topAncestor) { 24             depth++; 25             descendant = descendant.ancestor; 26         } 27         return depth; 28     } 29 30     public static AncestralTree backtrackAncestralTree( 31         AncestralTree lowerDescendant, 32         AncestralTree higherDescendant, 33         int diff 34     ) { 35         while (diff &gt; 0) { 36             lowerDescendant = lowerDescendant.ancestor; 37             diff--; 38         } 39         while (lowerDescendant != higherDescendant) { 40             lowerDescendant = lowerDescendant.ancestor; 41             higherDescendant = higherDescendant.ancestor; 42         } 43         return lowerDescendant; 44     } 45 46     public class AncestralTree { 47         public char name; 48         public AncestralTree ancestor; 49 50         public AncestralTree(char name) { 51             this.name = name; 52             this.ancestor = null; 53         } 54 55         // This method is for testing only. 56         public void AddAsAncestor(AncestralTree[] descendants) { 57             foreach (AncestralTree descendant in descendants) { 58                 descendant.ancestor = this; 59             } 60         } 61     } 62 } 63</pre>	<pre>1 public class Program { 2     public static AncestralTree GetYoungestCommonAncestor( 3         AncestralTree topAncestor, 4         AncestralTree descendantOne, 5         AncestralTree descendantTwo 6     ) { 7         // Write your code here. 8         return null; 9     } 10 11     public class AncestralTree { 12         public char name; 13         public AncestralTree ancestor; 14 15         public AncestralTree(char name) { 16             this.name = name; 17             this.ancestor = null; 18         } 19 20         // This method is for testing only. 21         public void AddAsAncestor(AncestralTree[] descendants) { 22             foreach (AncestralTree descendant in descendants) { 23                 descendant.ancestor = this; 24             } 25         } 26     } 27 } 28</pre>	

**Run or submit code when you're ready.**