**First Common Ancestor**

Design an algorithm and write code to find the first common ancestor of two nodes in a binary tree. Avoid storing additional nodes in a data structure. NOTE: This is not necessarily a binary search tree.

Solution**:**

public TreeNode commonAncestor**(**TreeNode root**,** TreeNode a**,** TreeNode b**){**

**if(!**cover**(**root **,** a**)||** **!**cover**(**root **,** b**)** **||** root **==** **null)**

**return** **null;**

**return** anscestorHelper**(**root**,** a**,** b**);**

**}**

public TreeNode anscestorHelper**(**TreeNode root**,** TreeNode a**,** TreeNode b**){**

**if(**root **==** **null** **||** root **==** a **||** root **==** b**)**

**return** root**;**

boolean isLeft\_a **=** cover**(**a**);**

boolean isLeft\_b **=** cover**(**b**);**

**if(**isLeft\_a **!=** isLeft**){**

**return** root**;**

**}**

TreeNode ancestor **=** isLeft\_a **?** root**.**left**:**root**.**right**;**

**return** anscestorHelper**(**root**,**a**,**b**);**

**}**

public boolean cover**(**TreeNode root**,**TreeNode a**){**

**if(**root **==** **null)**

**return** **false**

**return** **(**root **==** a **||** cover**(**root**.**right**)** **||** cover**(**root**.**left**))**

**}**