Our Solution(s) Run Code

Solution 1

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
  // O(d) time | O(1) space - where d is the depth (height) of the ancestral tree
  public static AncestralTree getYoungestCommonAncestor(
      AncestralTree topAncestor, AncestralTree descendantOne, AncestralTree descer
    int depthOne = getDescendantDepth(descendantOne, topAncestor);
    int depthTwo = getDescendantDepth(descendantTwo, topAncestor);
    if (depthOne > depthTwo) {
      return backtrackAncestralTree(descendantOne, descendantTwo, depthOne - depth
    } else {
      return backtrackAncestralTree(descendantTwo, descendantOne, depthTwo - depth
    }
  }

  public static int getDescendantDepth(AncestralTree descendant, AncestralTree top
    int depth = 0;
    while (descendant != topAncestor) {
      depth++;
      descendant = descendant.ancestor;
    }
    return depth;
  }

  public static AncestralTree backtrackAncestralTree(
      AncestralTree lowerDescendant, AncestralTree higherDescendant, int diff) {
    while (diff > 0) {
      lowerDescendant = lowerDescendant.ancestor;
      diff--;
    }
    while (lowerDescendant != higherDescendant) {
      lowerDescendant = lowerDescendant.ancestor;
      higherDescendant = higherDescendant.ancestor;
    }
    return lowerDescendant;
  }

  static class AncestralTree {
    public char name;
    public AncestralTree ancestor;

    AncestralTree(char name) {
      this.name = name;
      this.ancestor = null;
    }

    // This method is for testing only.
    void addAsAncestor(AncestralTree[] descendants) {
      for (AncestralTree descendant : descendants) {
        descendant.ancestor = this;
      }
    }
  }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
class Program {
  public static AncestralTree getYoungestCommonAncestor(
      AncestralTree topAncestor, AncestralTree descendantOne, AncestralTree descer
    // Write your code here.
    return null;
  }

  static class AncestralTree {
    public char name;
    public AncestralTree ancestor;

    AncestralTree(char name) {
      this.name = name;
      this.ancestor = null;
    }

    // This method is for testing only.
    void addAsAncestor(AncestralTree[] descendants) {
      for (AncestralTree descendant : descendants) {
        descendant.ancestor = this;
      }
    }
  }
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.