**Subtree of another tree**

Given two non-empty binary trees **s** and **t**, check whether tree **t** has exactly the same structure and node values with a subtree of **s**. A subtree of **s** is a tree consists of a node in **s** and all of this node's descendants. The tree **s** could also be considered as a subtree of itself.

class Solution **{**

public boolean isSubtree**(**TreeNode s**,** TreeNode t**)** **{**

**if(**s**==null** **&&** t**==null)** **return** **true;**

**if(**s**==null)** **return** **false;**

**if(**t**==null)** **return** **true;**

boolean isEquals **=** **false;**

**if(**s**.**val **==** t**.**val**)** **{**

isEquals **=** equals**(**s**.**left**,** t**.**left**)** **&&** equals**(**s**.**right**,** t**.**right**);**

**}**

isEquals **=** isEquals **||** **(**isSubtree**(**s**.**left**,** t**)** **||** **(**isSubtree**(**s**.**right**,** t**)));**

**return** isEquals**;**

**}**

private boolean equals**(**TreeNode s**,** TreeNode t**)** **{**

**if(**s**==null** **&&** t**==null)** **return** **true;**

**if((**s**==null** **&&** t**!=null)** **||** **(**s**!=null** **&&** t**==null))** **return** **false;**

**if(**s**.**val **==** t**.**val**)** **{**

**return** equals**(**s**.**left**,** t**.**left**)** **&&** equals**(**s**.**right**,** t**.**right**);**

**}else** **return** **false;**

**}**

**}**