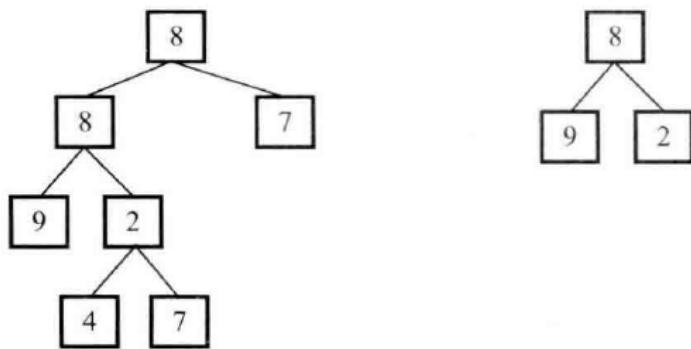


面试题 26：树的子结构

题目：输入两棵二叉树 A 和 B，判断 B 是不是 A 的子结构。二叉树节点的定义如下：

```
struct BinaryTreeNode
{
    double                m_dbValue;
    BinaryTreeNode*       m_pLeft;
    BinaryTreeNode*       m_pRight;
};
```

例如图 3.13 中的两棵二叉树，由于 A 中有一部分子树的结构和 B 是一样的，因此 B 是 A 的子结构。



解

递归：有三大种情况需要判断：

1. 包含A的根节点的树
2. A的左子树
3. A的右子树

每种情况中，如果和判断子树？

也是递归。递归的终止条件是：

1. 第二个空了，true
2. 第一个空了，第二个没空，false

根节点相等时，左孩子和右孩子对应节点必须相等

```
bool isSubtree(TreeNode* a,TreeNode* b)
{
    if(!a || !b)
        return false;
    return help(a,b) || isSubtree(a->left,b) || isSubtree(a->right,b);
}
bool help(TreeNode* p1,TreeNode* p2)
```

```
{
    if(!p2)
        return true;
    if(!p1)
        return false;
    if(p1->val == p2->val)
        return help(p1->left,p2->left) && help(p1->right,p2->right);
    else
        return false;
}
```