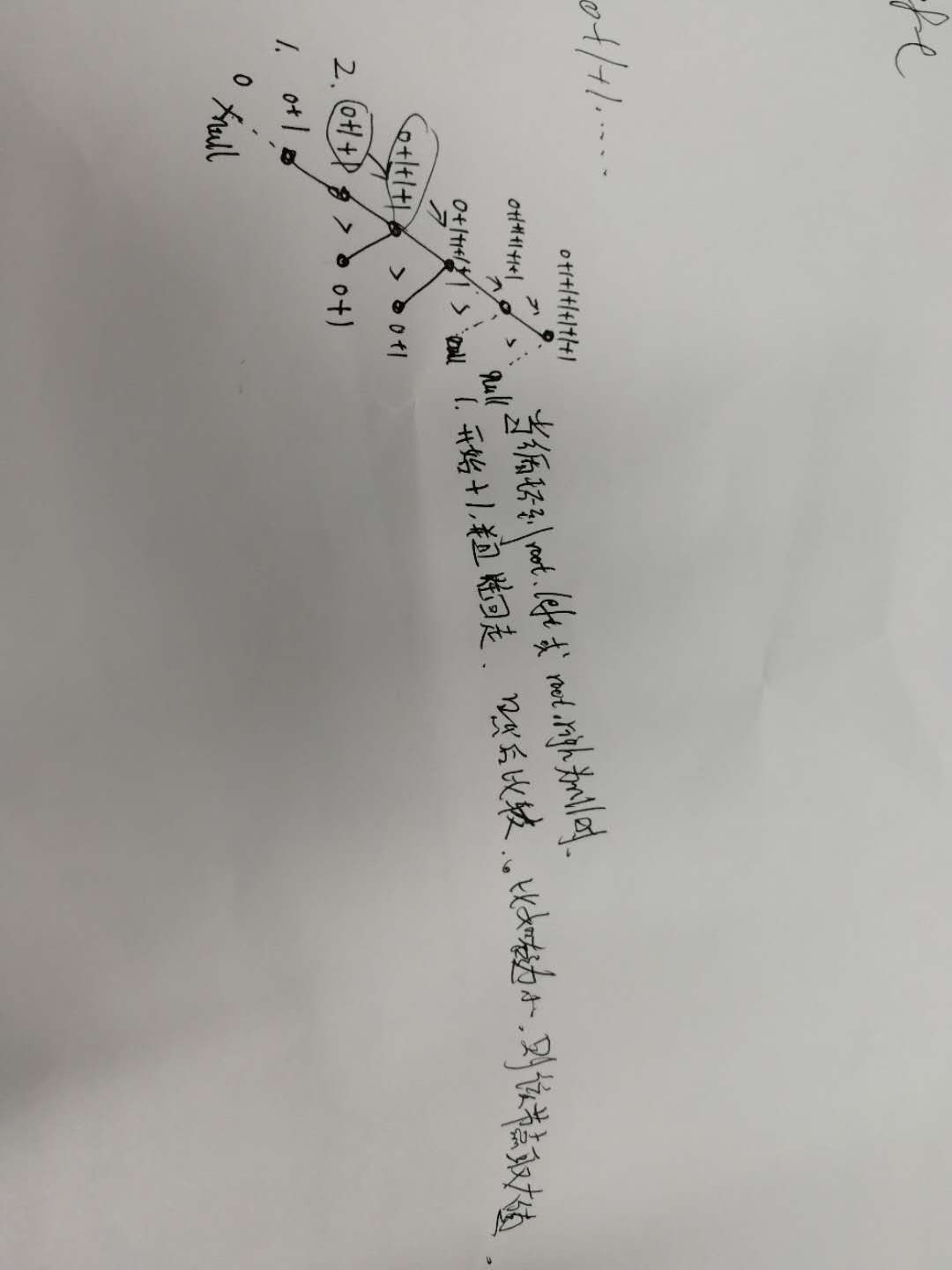
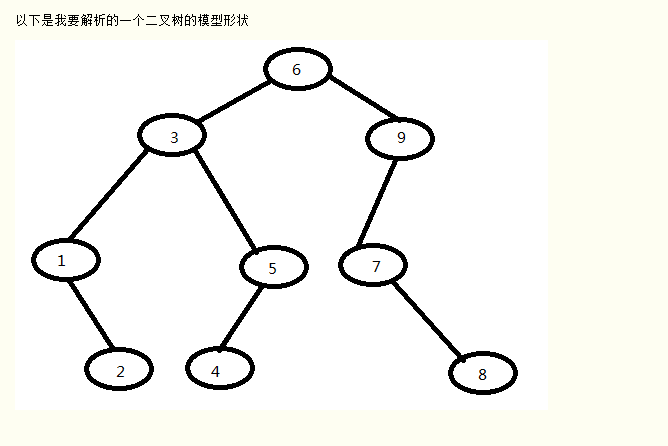
**题目描述**

输入一棵二叉树，求该树的深度。从根结点到叶结点依次经过的结点（含根、叶结点）形成树的一条路径，最长路径的长度为树的深度。





**package** facehandjava.tree;  
  
**public class** DepthTree {  
 **public static** Node init() {*//注意必须逆序建立，先建立子节点，再逆序往上建立，因为非叶子结点会使用到下面的节点，而初始化是按顺序初始化的，不逆序建立会报错* Node J = **new** Node(8, **null**, **null**);  
 Node H = **new** Node(4, **null**, **null**);  
 Node G = **new** Node(2, **null**, **null**);  
 Node F = **new** Node(7, **null**, J);  
 Node E = **new** Node(5, H, **null**);  
 Node D = **new** Node(1, **null**, G);  
 Node C = **new** Node(9, F, **null**);  
 Node B = **new** Node(3, D, E);  
 Node A = **new** Node(6, B, C);  
 **return** A; *//返回根节点* }  
  
 **public static void** main(String[] args) {  
*// L2RRecursiveBinaryTree tree = new L2RRecursiveBinaryTree();* Node root = DepthTree.*init*();  
 System.***out***.println(**"树的深度"**);  
 **int** L = *DepthTree*(root);  
 System.***out***.println(L);  
 }  
  
 **public static int** DepthTree(Node node) {  
 **if** (node == **null**) {  
 **return** 0;  
 }  
 **int** l = *DepthTree*(node.getLeftNode());  
 **int** r = *DepthTree*(node.getRightNode());  
 **int** d = l > r ? l : r;  
 **return** d+1;  
 }  
}