**题目描述**

输入两个链表，找出它们的第一个公共结点。

两种思路，个人喜欢第一种：

1、L1 l2各自走一遍，如果都到底部还未相遇，则证明l1 l2长度不一样，分别到null后，l1换l2链表头开始走，l2换l1链表头走，然后接着他们就会相遇了，遇到的第一个点就是公共点。因为大家所走的路是相同的。

2、用一个stact栈存2个链表的数据，然后开始pop，pop到不相等的，前1个想等的为他们的公共节点。所以需要用到额外的存储空间。

下面只给出我第一种思路自己写的程序：

**package** facehandjava.Linked;  
  
**public class** SameNode {  
 **public static void** main(String[] args) {  
  
 Node n6 = **new** Node(11,**null**);  
 Node n5 = **new** Node(9, n6);  
 Node n4 = **new** Node(7, n5);  
 *//共同节点是n4* Node n3 = **new** Node(5, n4);  
 Node n2 = **new** Node(3, n3);  
 Node n1 = **new** Node(1, n2);  
  
 Node n33 = **new** Node(22, n4);  
 Node n22 = **new** Node(22, n33);  
 Node n11 = **new** Node(11, n4);  
  
 Node sameNode = *SameNode*(n1, n11);  
 System.***out***.println(sameNode.getVal());  
 }  
  
 **public static** Node SameNode(Node n1, Node n2) {  
 Node node1 = n1;  
 Node node2 = n2;  
 **while** (node1 != node2) {  
 **if** (node1 != **null** && node2 != **null**) {  
 node1 = node1.getNext();  
 node2 = node2.getNext();  
 **if** (node1 == node2) {  
 **return** node1;  
 }  
 }  
 **if** (node1 == **null**) {  
 node1 = n2;  
 node2 = node2.getNext();  
 }  
 **if** (node2 == **null**) {  
 node2 = n1;  
 node1 = node1.getNext();  
 }  
 }  
 **return** node1;  
 }  
}