100 相同的树

Label：二叉树
给你两棵二叉树的根节点 p 和 q ，编写一个函数来检验这两棵树是否相同。
如果两个树在结构上相同，并且节点具有相同的值，则认为它们是相同的。

- 递归

```
class Solution {
    public boolean isSameTree(TreeNode p, TreeNode q) {
        if (p == null && q == null) {
            return true;
        }else if (p == null ^ q == null) {
            return false;
        }
        if (p.val != q.val)  return false;
        return isSameTree(p.right, q.right) && isSameTree(p.left, q.left);
    }
}
```

- 广度优先

```
class Solution {
    public boolean isSameTree(TreeNode p, TreeNode q) {
        if (p == null && q == null) {
            return true;
        } else if (p == null || q == null) {
            return false;
        }
        Queue<TreeNode> queue1 = new LinkedList<TreeNode>();
        Queue<TreeNode> queue2 = new LinkedList<TreeNode>();
        queue1.offer(p);
        queue2.offer(q);
        while (!queue1.isEmpty() && !queue2.isEmpty()) {
            TreeNode node1 = queue1.poll();
            TreeNode node2 = queue2.poll();
            if (node1.val != node2.val)  return false;
            TreeNode left1 = node1.left, right1 = node1.right, left2 =
node2.left, right2 = node2.right;
            if (left1 == null ^ left2 == null)  return false;// 异或
            if (right1 == null ^ right2 == null)  return false;

            if (left1 != null)  queue1.offer(left1);
            if (right1 != null)  queue1.offer(right1);
            if (left2 != null) queue2.offer(left2);
            if (right2 != null) queue2.offer(right2);
        }
        return queue1.isEmpty() && queue2.isEmpty();
    }
}
```