

```

/**
 * 三种写法查找二分查找树是否含有某个节点
 * @param binarySearchTree 二分查找树
 * @param s 节点数据
 * @return boolean
 */
private static boolean lookup(BinarySearchTree
binarySearchTree, char s){
    //ifelse形式
    //    if(null==binarySearchTree){
    //        return false;
    //    } else {
    //        if(s==binarySearchTree.data) {
    //            return true;
    //        } else if(s<binarySearchTree.data){
    //            return lookup(binarySearchTree.leftSubTree, s);
    //        } else {
    //            return lookup(binarySearchTree.rightSubTree, s);
    //        }
    //    }
    //逻辑运算符形式
    //    return (null!=binarySearchTree)
    //        && (
    //            (s==binarySearchTree.data) ||
    //            (
    //                (s<binarySearchTree.data &&
    lookup(binarySearchTree.leftSubTree, s)) ||
    //                (s>binarySearchTree.data &&
    lookup(binarySearchTree.rightSubTree, s))
    //            )
    //        );
    //三木运算符形式
    return (null==binarySearchTree)?false:
        ((s==binarySearchTree.data)?true:
            (s<binarySearchTree.data)?
lookup(binarySearchTree.leftSubTree, s):
lookup(binarySearchTree.rightSubTree, s)

```

