Answers

1.
In a binary tree, every node has at most two children. There is no ordering of values imposed on a binary tree.

A binary search tree is a binary tree with ordering imposed on the values of the nodes. All the values in the nodes on the left side of the tree are smaller than the root, and all the values in the nodes on the right side are larger than the root.

2.

```
def same_shape(t1, t2):
    if t1 == None and t2 == None:
        return True
    if (t1 == None and t2 != None) or (t1 != None and t2 == None):
        return False
   return same_shape(t1._left, t2._left) and same_shape(t1._right, t2._right)
```

3.

```
def concat(alist, blist):

    #ensure that alist is not empty
    if alist._head == None:
        return blist

    #avoid traversing alist if blist is empty
    if blist._head == None:
        return alist

    curr = alist._head
    while curr._next != None:
        curr = curr._next

    curr._next = blist._head
    return alist
```