Given a binary tree, determine if it is a valid binary search tree (BST).

Assume a BST is defined as follows:

* The left subtree of a node contains only nodes with keys **less than** the node's key.
* The right subtree of a node contains only nodes with keys **greater than** the node's key.
* Both the left and right subtrees must also be binary search trees.

思路1：

1. 先写函数分别获取该树的最左边节点（二叉搜索树中最小的节点）和最右节点（二叉搜索树中最大节点）
2. 判断根节点的左子树中最右的节点是否小于根节点，接着递归判断根节点的左子树是否是二叉搜索树，如果同时满足则根节点的左子树是二叉搜索树。
3. 同理。判断根节点的右子树中最左的节点是否大于根节点，递归判断右子树是否是二叉搜索树，同时满足则右子树是二叉搜索树。
4. 如果都是二叉搜索树，则返回Ture