1008. Construct Binary Search Tree from Preorder Traversal

Return the root node of a binary **search** tree that matches the given `preorder` traversal.

(Recall that a binary search tree is a binary tree where for every node, any descendant of `node.left` has a value `<` `node.val`, and any descendant of `node.right` has a value `>` `node.val`. Also recall that a preorder traversal displays the value of the `node` first, then traverses `node.left`, then traverses `node.right`.)

```
class Solution {
    int i = 0;
    public TreeNode bstFromPreorder(int[] preorder) {

        // Give the function a bound the maximum number it will handle.
        // The left recursion will take the elements smaller than node.val
        // The right recursion will take the remaining elements smaller than
bound

        return bstFromPreOrder(preorder, Integer.MAX_VALUE);
    }

    private TreeNode bstFromPreOrder(int[] preOrder, int bound) {
        if (i == preOrder.length || preOrder[i] > bound)
            return null;

        TreeNode root = new TreeNode(preOrder[i++]);
        root.left = bstFromPreOrder(preOrder, root.val);
        root.right = bstFromPreOrder(preOrder, bound);
        return root;
    }
}
```