```c
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */
#define ALLOC_LENGTH   (100)
int* result;
int alloc_length;
#if 1
bool _isValidBST(struct TreeNode* root, long mn, long mx)
{
        if (!root) return true;
        if (root->val <= mn || root->val >= mx) return false;
        return _isValidBST(root->left, mn, root->val) && _isValidBST(root->right, root->val, mx);
}

bool isValidBST(struct TreeNode* root) {

    return _isValidBST(root, LONG_MIN, LONG_MAX);
}


#else
void buildBSTList(struct TreeNode* root, int* result_index)
{
    if(root)
    {
        buildBSTList(root->left, result_index);
        result[*result_index] = root->val;
        (*result_index)++;
        if( 0 == (*result_index % ALLOC_LENGTH) )
        {
            alloc_length += ALLOC_LENGTH;
            result = (int*)realloc(result, sizeof(int)*alloc_length);
        }
        buildBSTList(root->right, result_index);
    }
}

bool isValidBST(struct TreeNode* root) {
    int result_index;
    int index;

    result_index = 0;
    alloc_length = ALLOC_LENGTH;
    result = (int*)malloc(sizeof(int)*alloc_length);

    buildBSTList(root, &result_index);

    for(index = 1 ; index < result_index; index++)
    {
        if(result[index - 1] >= result[index])
        {
            return false;
        }
    }

    return true;
}
#endif
```