```
/*
给定二叉搜索树（BST）的根节点和一个值。 你需要在BST中找到节点值等于给定值的节点。 返回以该节点为根的子树。 如果节点不存在，则返回 NULL。

例如，

给定二叉搜索树:

        4
       / \
      2   7
     / \
    1   3

和值: 2

你应该返回如下子树:

      2
     / \
    1   3

在上述示例中，如果要找的值是 5，但因为没有节点值为 5，我们应该返回 NULL。

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/search-in-a-binary-search-tree
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析:
只要找到$value = val$的节点即可

- 方法一:递归法
- 方法二:迭代法

方法一:C_递归法

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */



struct TreeNode* ret_val = NULL;


void __searchBSTNode(struct TreeNode* node, int val)
{
```

```
    if(node==NULL)
    {
        return;
    }

    if (node->val == val)
    {
        ret_val = node;
        return;
    }
    else if (node->val < val)
    {
        __searchBSTNode(node->right,val);
    }
    else
    {
        __searchBSTNode(node->left,val);
    }


}

struct TreeNode* searchBST(struct TreeNode* root, int val)
{
    ret_val = NULL;
    __searchBSTNode(root,val);
    return ret_val;
}

/*
执行结果:
通过
显示详情
执行用时 :64 ms, 在所有 C 提交中击败了9.70% 的用户
内存消耗 :16.4 MB, 在所有 C 提交中击败了100.00%的用户
*/
```

方法二:C_迭代法

```
struct TreeNode* searchBST(struct TreeNode* root, int val)
{

    struct TreeNode* temp = root;
    while (1)
    {
        if(temp==NULL)
        {
            break;
        }

        if(temp->val< val)
        {
            temp = temp -> right;
        }
        else if(temp->val > val)
        {
```

```
            temp = temp -> left;
        }
        else
        {
            return temp;
        }
    }
    return NULL;
}

/*
执行结果:
通过
显示详情
执行用时 :56 ms, 在所有 C 提交中击败了14.93% 的用户
内存消耗 :16.4 MB, 在所有 C 提交中击败了100.00%的用户
*/



struct TreeNode* searchBST(struct TreeNode* root, int val)
{
    while (1)
    {
        if(root==NULL)
        {
            break;
        }
        if(root->val< val)
        {
            root = root -> right;
        }
        else if(root->val > val)
        {
            root = root -> left;
        }
        else
        {
            return root;
        }
    }
    return NULL;
}

/*
执行结果:
通过
显示详情
执行用时 :36 ms, 在所有 C 提交中击败了68.66% 的用户
内存消耗 :16.6 MB, 在所有 C 提交中击败了100.00%的用户
*/
```

AlimyBreak
2019.09.06