/\*\*

\* Definition for a binary tree node.

\* function TreeNode(val) {

\* this.val = val;

\* this.left = this.right = null;

\* }

\*/

/\*\*

\* @param {number} n

\* @return {TreeNode[]}

\*/

/\* 主要思路： 利用递归算法，遍历1~n之间的各个值作为根结点建立树，对于i为根的树，1~i-1肯定为其左子树上的节点，i+1~n必为其右子树上的节点，然后用左子树集合和右子树集合分别与根节点组合（因此左右子树集合的组合 两个for循环放在根节点i的for 循环之中），要注意由于返回的是结构体数组所以遍历的时候用 for（let i of set）,i代表集合中的一个元素也即一颗子树 \*/

var maketree = function(left,right)

{

let res = [];

if(left>right)

{

res.push(null);

return res;

}

for(let i = left;i <= right;i++)

{

let lefttree = maketree(left,i-1);

let righttree = maketree(i+1,right);

for(const l of lefttree) //要注意这里

for(const r of righttree)

{

const root = new TreeNode(i);

root.left = l;

root.right = r;

res.push(root);

}

}

return res;

};

var generateTrees = function(n) {

if(n==0)

return [];

return maketree(1,n);

};