二叉查找树：根节点左子树的值全部小于根节点，右子树的值全部大于根节点

二叉查找树具有的性质：二叉搜索树的中序满足是一组序列的从小到大排列，所以只需排序所给序列即可得到中序

根据完全搜索树的结点数，可以求出它的根结点在中序中对应的下标

1. 数据结构

struct node{

int v;

node\* leftchild=NULL;

node\* rightchild=NULL;

} t; 树根节点

1. 算法

用树的遍历序列构建BST

for (int i = 1; i < n; i++)

{

node\* tem = (node\*)malloc(sizeof(node)); //#include<memory.h>

tem->leftchild = tem->rightchild = NULL; //初始化，定义结构体时初始化也可以

tem->v = s[i];

int flag = 1;

node\* u = &t;

while (flag)

{

if (tem->v>=u->v)

{

if (u->rightchild==NULL)

{

u->rightchild = tem;

flag = 0;

}

else

{

u = u->rightchild;

}

}

if (tem->v<u->v)

{

if (u->leftchild==NULL)

{

u->leftchild = tem;

flag = 0;

}

else

{

u = u->leftchild;

}

}

}

}

求镜像树

void reverse(node\* t){

if (t->rightchild!=NULL||t->rightchild!=NULL)

{

node\* tem;

tem = t->leftchild;

t->leftchild = t->rightchild;

t->rightchild = tem;

}

if (t->rightchild!=NULL)

{

reverse(t->rightchild);

}

if (t->leftchild!=NULL)

{

reverse(t->leftchild);

}

}