struct node {

int num;

int lc;

int rc;

node()

{

this->lc = 0;

this->rc = 0;

}

};

vector<int>in;

vector<node>sto;

vector<int>pre;

int cnt = 1;

int cons(int in\_l, int in\_r, int pre\_l, int pre\_r)

{

if (pre\_l == pre\_r)

{

sto[cnt].num = pre[pre\_l];

return cnt++;

}

int tar = pre[pre\_l];

int tar\_in;

for(int i=in\_l;i<in\_r + 1; i++)

if (in[i] == tar){

tar\_in = i;

break;

}

int cur = cnt++;

sto[cur].num = tar;

if(tar\_in!=in\_l)

sto[cur].lc = cons(in\_l, tar\_in - 1, pre\_l + 1, pre\_l + 1 + tar\_in - in\_l - 1);//若存在，则构建左子树

if(tar\_in!=in\_r)

sto[cur].rc = cons(tar\_in + 1, in\_r, pre\_l + 1 + tar\_in - in\_l, pre\_r);//若存在，则构建右子树

return cur;

}