给你一个前序遍历和中序遍历，要求后序。

可以由先序和中序的性质得到 ： 先序的第一个借点肯定是当前子树的根借点， 那么在

中序中找到这个结点， 则这个结点左边的节点属于左子树， 右边的属于右子树。然后递归遍历就可以了。

注意这里的定义不能用char而需要用int

因为这不停的Output Limit Exceeded 和 RE

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

stack<int> \_stack;

int s1[1010],s2[1010];

int n;

void build(int l1,int r1,int l2,int r2)

{

if(l1>r1 || l2>r2)return;

int root=s1[l1];

\_stack.push(root);

int pos(-1);

int i(-1);

for(i=l2; i<=r2; i++)

if(s2[i]==root)

break;

int j=l1+i-l2+1;

if(j<=r1 && i+1<=r2)

build(j,r1,i+1,r2);

if(l1+1<=j-1 && l2<=i-1)

build(l1+1,j-1,l2,i-1);

}

int main()

{

//freopen("input.txt","r",stdin);

while(cin>>n)

{

for(int i=0; i<n; i++)

cin>>s1[i];

for(int i=0; i<n; i++)

cin>>s2[i];

build(0,n-1,0,n-1);

while(!\_stack.empty())

{

cout<<\_stack.top();

\_stack.pop();

if(!\_stack.empty())

cout<<' ';

}

puts("");

}

return 0;

}

#include<iostream>

#include<cstdio>

#include<cstring>

#include<stack>

using namespace std;

const int N=1010;

int n,pre[N],in[N]; //先序数组和后序数组

stack<int> st; //存放父节点

void build(int l1,int r1,int l2,int r2)

{ //l1,r1,是先序遍历的数组的开始和末尾，l2,r2是中序遍历的数组的开始和末尾

int i,j;

st.push(pre[l1]); //父节点入栈

for(i=l2;i<=r2;i++)

if(in[i]==pre[l1]) //找到父节点在中序遍历的位置i

break;

j=l1+(i-l2+1); //确定左子树和右子树在先序遍历的分界点j，即右子树的父节点

if(j<=r1 && i+1<=r2) //求解右子树

build(j,r1,i+1,r2);

if(l1+1<=j-1 && l2<=i-1) //求解左子树

build(l1+1,j-1,l2,i-1);

}

int main(){

//freopen("input.txt","r",stdin);

while(~scanf("%d",&n))

{

for(int i=0;i<n;i++)

scanf("%d",&pre[i]);

for(int i=0;i<n;i++)

scanf("%d",&in[i]);

build(0,n-1,0,n-1);

while(!st.empty())

{

printf("%d",st.top());

st.pop();

if(!st.empty())

printf(" ");

}

printf("\n");

}

return 0;

}