**算法训练 FBI树**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　我们可以把由“0”和“1”组成的字符串分为三类：全“0”串称为B串，全“1”串称为I串，既含“0”又含“1”的串则称为F串。  
　　FBI树是一种二叉树，它的结点类型也包括F结点，B结点和I结点三种。由一个长度为2N的“01”串S可以构造出一棵FBI树T，递归的构造方法如下：  
　　1)T的根结点为R，其类型与串S的类型相同；  
　　2)若串S的长度大于1，将串S从中间分开，分为等长的左右子串S1和S2；由左子串S1构造R的左子树T1，由右子串S2构造R的右子树T2。  
　　现在给定一个长度为2N的“01”串，请用上述构造方法构造出一棵FBI树，并输出它的后序遍历序列。

输入格式

　　第一行是一个整数N（0 <= N <= 10），第二行是一个长度为2N的“01”串。

输出格式

　　包括一行，这一行只包含一个字符串，即FBI树的后序遍历序列。

样例输入

3  
10001011

样例输出

IBFBBBFIBFIIIFF

数据规模和约定

　　对于40%的数据，N <= 2；  
　　对于全部的数据，N <= 10。  
　　注：  
　　[1] 二叉树：二叉树是结点的有限集合，这个集合或为空集，或由一个根结点和两棵不相交的二叉树组成。这两棵不相交的二叉树分别称为这个根结点的左子树和右子树。  
　　[2] 后序遍历：后序遍历是深度优先遍历二叉树的一种方法，它的递归定义是：先后序遍历左子树，再后序遍历右子树，最后访问根。

锦囊1

二叉树。

锦囊2

使用题设的方法直接构造一棵二叉树，然后再遍历。

本题的C++参考代码如下：

#include<cstdio>

#include<cstdlib>

#include<cstring>

int a[2100][3],f[1100],v=1,s=1;

void g(int x,int y)

{

if(y-x+1>1)

{

int vv=v,v1,v2;

a[vv][0]=v+1;v++;v1=v;

g(x,x+(y-x)/2);

a[vv][1]=v+1;v++;v2=v;

g(x+(y-x)/2+1,y);

if(a[v1][2]==a[v2][2])a[vv][2]=a[v1][2];

else a[vv][2]=2;

}

else{a[v][0]=-1;a[v][1]=-1;a[v][2]=f[s];s++;}

}

void h(int x)

{

if(a[x][0]!=-1)h(a[x][0]);

if(a[x][1]!=-1)h(a[x][1]);

if(a[x][2]==0)printf("B");

if(a[x][2]==1)printf("I");

if(a[x][2]==2)printf("F");

}

int main()

{

int n=1,i,nn;

char c;

scanf("%d\n",&nn);

for(i=1;i<=nn;i++)n\*=2;

for(i=1;i<=n;i++){scanf("%c",&c);f[i]=c-'0';}

g(1,n);

h(1);

printf("\n");

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

#include <malloc.h>

char in[1025];

typedef struct node

{

char data;

struct node \*lchild,\*rchild;

}Binode;

Binode \*create(int a,int b)

{

char ch;

Binode \*p;

int i,count0=0,count1=0,flag=0;

// scanf("%c",&ch);

// getchar();

if(a==b)

{

(in[a]=='0')?(ch='B'):(ch='I');

p=(Binode \*)malloc(sizeof(Binode));

p->data=ch;

p->lchild=NULL;

p->rchild=NULL;

return p;

}

for(i=a;i<=b;i++)

{

(in[i]=='0')?(count0++):(count1++);

if(count0 && count1)

{

ch='F';

flag=1;

break;

}

}

if(!flag)

{

if(!count0)

ch='I';

else

ch='B';

}

p=(Binode \*)malloc(sizeof(Binode));

p->data=ch;

p->lchild=create(a,((b+1)-a)/2+a-1);

p->rchild=create(((b+1)-a)/2+a,b);

return p;

}

void display(Binode \*p)

{

if(!p)

return;

display(p->lchild);

display(p->rchild);

printf("%c",p->data);

}

int main()

{

Binode \*r;

char c;

int p=0,n;

scanf("%d",&n);

getchar();

while(1)

{

c=getchar();

if(c=='\n')

break;

in[p++]=c;

}

r=create(0,p-1);

display(r);

return 0;

}

本题的Java参考代码如下：

import java.util.\*;

import java.lang.\*;

public class Main {

static class TreeNode

{

String data;

TreeNode lchild ;

TreeNode rchild ;

public String getData()

{

return data;

}

public TreeNode getLchild()

{

return lchild;

}

public TreeNode getRchild()

{

return rchild;

}

public void setNode(String data,TreeNode lc,TreeNode rc){

this.data = data;

lchild = lc;

rchild = rc;

}

}

static TreeNode create(TreeNode root,String ss)

{

if(ss.length()==1)

{

root.setNode(ss, null, null);

return root;

}

TreeNode tl=new TreeNode();

TreeNode tr=new TreeNode();

root.setNode(ss, create(tl,ss.substring(0,ss.length()/2)), create(tr,ss.substring(ss.length()/2,ss.length())));

return root;

}

static void ff(TreeNode root)

{

if(root!=null)

{

ff(root.getLchild());

ff(root.getRchild());

System.out.print(ff1(root.getData()));

}

}

static char ff1(String s)

{

int i,j;

char c='F';

for(i=0;i<s.length();i++)

if(s.charAt(i)=='1')

break;

for(j=0;j<s.length();j++)

if(s.charAt(j)=='0')

break;

if(i==s.length())

c='B';

if(j==s.length())

c='I';

return c;

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

String s=sc.next();

sc.close();

int m=1,j;

TreeNode root=new TreeNode();

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

m\*=2;

for(j=0;j<s.length();j++)

if(s.charAt(j)>'1'||s.charAt(j)<'0')

break;

if(s.length()!=m||j<s.length())

System.out.print("error");

else

{

create(root,s);

ff(root);

}

}

}