#include<iostream>

#include<string>

using namespace std;

class HuffTree {

public:

string data;

int weight;

int parent, lchild, rchild;

};

void select(HuffTree ht[], int& s1, int& s2, int n) {

s1 = 0;

s2 = 0;

int temp = 1000;

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

if (temp > ht[i].weight && ht[i].parent == -1)

temp = ht[i].weight;

s1 = i;

}

temp = 1000;

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

if (temp > ht[i].weight && ht[i].parent == -1 && i != s1) {

temp = ht[i].weight;

s2 = i;

}

}

}

void creat(HuffTree ht[],int n) {

int i = 0;

int m;

m = 2 \* n-1;

int s1, s2;

for (i = 0; i < m; i++) {

ht[i].parent = -1;

ht[i].lchild = -1;

ht[i].rchild = -1;

}

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

cout << "请输入测试字符" << endl;

cin >> ht[i].data;

cout << "请输入节点的权值" << endl;

cin >> ht[i].weight;

}

for (i = n; i < m; i++) {

select(ht, s1, s2,n);

ht[s1].parent = i;

ht[s2].parent = i;

ht[i].lchild = s1;

ht[i].lchild = s2;

ht[i].weight = ht[s1].weight + ht[s2].weight;

}

}

void haffcode(HuffTree ht[], string haff[], int n) {

int temp;

int parent;

int start;

//遍历哈夫曼树，生成哈夫曼编码

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

temp = i;

parent = ht[i].parent;

while (parent != -1) {

if (ht[parent].lchild == temp)

{

haff[i] = haff[i] + '0';

cout << "CESJHI" << haff[i]<<endl;

}

else

{

haff[i] = haff[i] + '1';

cout << "CESJHI" << haff[i] << endl;

}

temp = parent;

parent = ht[parent].parent;

}

}

}

int main(){

HuffTree ht[5];

string haff[5];

creat(ht,5);

haffcode(ht, haff, 5);

cout << "weight parent left right" << endl;

for (int i = 0; i < 2 \* 5 - 1; i++)

cout << ht[i].weight << " " << ht[i].parent << " "

<< ht[i].lchild << " " << ht[i].rchild << endl;

for (int i = 0; i < 5; i++) {

cout << haff[i] << " " ;

}

system("pause");

return 0;

}

