代码：（截图再代码后面）

#include <iostream>

#include <queue>

using namespace std;

typedef struct E

{

int number;

int w;

struct E \*next;

} Bian;

typedef struct V//顶点

{

int number;

Bian \*first;

} Ding;

class Linjie

{

private:

Ding \*Table; //指向邻接表的指针

int ding\_num; //定点数量

int bian\_num; //边数量

int \*help; //用于遍历的辅助数组指针

void Find\_DFS(int number);

public:

Linjie(int ding = 0 , int bian = 0); //初始化邻接表，传入参数分别为顶点数和边数

void Create(); //创建邻接表

void Print(); //打印邻接表

void DFS(); //深度优先遍历

void BFS(); //广度优先遍历

};

Linjie::Linjie(int ding , int bian)

{

this->ding\_num = ding;

this->bian\_num = bian;

this->Table = new Ding[ding];

this->help = new int[ding];

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

{

help[i] = 0;

}

}

void Linjie::Create()

{

for(int i = 0 ; i < this->ding\_num ; ++i) //初始化各个顶点的链表头

{

this->Table[i].number = i+1;

this->Table[i].first = NULL;

}

int a = 0,b = 0;

for(int i = 0 ; i < this->bian\_num ; ++i)

{

cout << "请输入一条边两端顶点的序号，空格隔开" << endl;

cin >> a >> b;

/\*1\*/

Bian \*t1 = new Bian; //创建临时的边

t1->number = b; //边的其中一个顶点的序号

t1->w = 1; //边的权值

t1->next = this->Table[a-1].first; //头插

this->Table[a-1].first = t1; //头插

/\*2\*///重复1但是换边了

Bian \*t2 = new Bian;

t2->number = a;

t2->w = 1;

t2->next = this->Table[b-1].first;

this->Table[b-1].first = t2;

}

}

void Linjie::Print()

{

for(int i = 0 ; i < this->ding\_num ; ++i)

{

cout << (i+1);

for(Bian \*t = this->Table[i].first ; t != NULL ; t = t->next)

{

cout << "->" << t->number;

}

cout << endl;

}

}

void Linjie::Find\_DFS(int number)

{

this->help[number] = 1;

cout << this->Table[number].number << " ";

Bian \*t = this->Table[number].first;

if(this->help[t->number-1] == 0)

{

this->Find\_DFS(t->number-1);

}

}

void Linjie::DFS() //深度优先遍历

{

for(int i = 0 ; i < this->ding\_num ; ++i)

{

help[i] = 0;

}

for(int i = 0 ; i < this->ding\_num ; ++i)

{

if(this->help[i] == 0)

{

this->Find\_DFS(i);

}

}

}

void Linjie::BFS() //广度优先遍历

{

queue<int> Que;

for(int i = 0 ; i < this->ding\_num ; ++i)

{

help[i] = 0;

}

for(int i = 0 ; i < this->ding\_num ; ++i)

{

if(this->help[i] == 0)

{

this->help[i] = 1;

cout << this->Table[i].number << " ";

Que.push(i);

while(Que.empty() == true)

{

int index = 0;

index = Que.front();

Bian \*t = this->Table[index].first;

while(t)

{

if(this->help[t->number-1] == 0)

{

this->help[t->number-1] = 1;

cout << t->number << " ";

Que.push(t->number-1);

t = t->next;

}

}

}

}

}

}

int main()

{

Linjie a(6 , 6);

a.Create();

a.Print();

cout << "深度优先遍历：" << endl;

a.DFS();

cout << endl;

cout << "广度优先遍历：" << endl;

a.BFS();

}

截图：

