#include<stdio.h>

#include<malloc.h>

#define MAXV 100

int A[MAXV][MAXV] = {

{0,0,1,0},

{0,0,0,1},

{1,0,0,1},

{0,1,1,0}

};

typedef struct ANode

{

int adjvex;//邻接点编号

struct ANode\* nextarc;

int weight;

}ArcNode;//结点的类型

typedef struct Vnode

{

ArcNode\* firstarc;

}VNode;//邻接表的头结点

typedef struct

{

VNode adjlist[MAXV];//头结点数组

int n;//顶点个数

int e;//边数

}AdjGraph;//完整的图

//创建图

void CreateAdj(AdjGraph\*& G, int A[MAXV][MAXV], int n, int e)

{

G = (AdjGraph\*)malloc(sizeof(AdjGraph));

ArcNode\* p;

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

{

G->adjlist[i].firstarc = NULL;

}

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

{

for (int j = n - 1;j >= 0;j--)

{

if (A[i][j] != 0)

{

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

p->adjvex = j;

p->nextarc = G->adjlist[i].firstarc;

G->adjlist[i].firstarc = p;

}

}

}

G->n = n;

G->e = e;

}

//图的遍历(广度优先)

int visited\_A[MAXV];

void BFS(AdjGraph\* G, int v)

{

visited\_A[MAXV] = { 0 };

int B[MAXV];

int front, rear;

front = rear = -1;

B[++rear] = v;

ArcNode\* p;

visited\_A[v] = 1;

printf("%d ", v);

while (front != rear)

{

p = G->adjlist[B[++front]].firstarc;

while (p != NULL)

{

if (visited\_A[p->adjvex] == 0)

{

printf("%d ", p->adjvex);

B[++rear] = p->adjvex;

visited\_A[p->adjvex] = 1;

}

p = p->nextarc;

}

}

printf("\n");

}

//图的遍历(深度优先)

int visited\_B[MAXV] = { 0 };

void DFS(AdjGraph\* G, int v)

{

if (visited\_B[v] == 0)

{

printf("%d ", v);

visited\_B[v] = 1;

}

ArcNode\* p;

p = G->adjlist[v].firstarc;

while (p != NULL)

{

if (visited\_B[p->adjvex] == 0)

{

DFS(G, p->adjvex);

}

p = p->nextarc;

}

}

//打印输出邻接表

void print(AdjGraph\* G)

{

printf("\n打印输出邻接表:\n");

ArcNode\* p;

for (int i = 0;i < G->n;i++)

{

p = G->adjlist[i].firstarc;

printf("%d", i);

while (p != NULL)

{

printf("->%d", p->adjvex);

p = p->nextarc;

}

printf("\n");

}

}

int main()

{

AdjGraph\* G;

CreateAdj(G, A, 4, 3);

printf("图的遍历(广度优先):\n");

BFS(G, 2);

printf("图的遍历(深度优先):\n");

DFS(G, 0);

print(G);

return 0;

}

