6-16

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#define N 5

#define E 6

int visited[N];

typedef int vextype;

typedef struct node{ //邻接链表结点

int adjvex;

struct node \*next;

}edgenode;

typedef struct{

vextype vertex; //顶点域

edgenode \*link; //指针域

}vexnode;

vexnode ga[N]; //顶点表 ,ga[i]表示V(i+1),从V1开始，无V0

void CreateAdjlist(vexnode ga[]){

int i,j,k;

edgenode \*s;

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

scanf("%d",&ga[i].vertex); //读入顶点信息

ga[i].link=NULL;

}

for(k=0;k<E;k++){

scanf("%d %d",&i,&j);

s=(edgenode\*)malloc(sizeof(edgenode));

s->adjvex=j;

s->next=ga[i-1].link;

ga[i-1].link=s;

}

}

int DFSL(int i,int j){ //深度优先算法，输入下标值，即判断[Vi,Vj]则输入[i-1,j-1]

edgenode \*p;

if(ga[i].vertex==ga[j].vertex){

return 1; //存在路径，返回1

}

visited[i]=1;

p=ga[i].link;

while(p!=NULL){

if(visited[p->adjvex-1]==0){

DFSL(p->adjvex-1,j);

if(DFSL) return 1; //存在路径，返回1

}

p=p->next;

}

return 0;

}

int main(){

int k=0,int m,n;

CreateAdjlist(ga);

scanf("%d %d",&m,&n);

DFSL(m,n); //判断ga[m]与ga[n]间是否存在路径

if(DFSL(m,n)) printf("Yes\n");

else printf("No\n");

return 0;

}

广度优先算法

int BFSL(int k,int l){

int i;

edgenode \*p;

SetNull(Q);

if(ga[k].vertex==ga[l].vertex){

return 1; //存在路径，返回1

}

visited[k]=1;

EnQueue(Q,k);

while(!Empty(Q)){

i=DeQueue(Q);

p=ga[i].link;

while(p!=null){

if(visited[p->adjvex]!=1){

if(ga[p->adjvex].vertex==ga[l].vertex){

return 1; //存在路径，返回1

}

visited[p->adjvex]=1;

EnQueue(Q,p->adjvex);

}

p=p->next;

}

}

return 0;

}