#include<stdio.h>

#include<conio.h>

int size=8;

int adj[8][8]={{0,40,60,0,0,0,0,0},{0,0,0,20,0,0,0,0},{0,0,0,10,15,0,0,0},{0,0,0,0,0,30,0,0},{0,0,0,0,0,0,10,0},{0,0,0,0,0,0,10,60},{0,0,0,0,0,0,0,50},{0,0,0,0,0,0,0,0}}; //directed weighted graph where weights signify mag of similarity

int flow[8][8];

int visited[8];

int sat[8][8];

int source[8];

int i,j,b,k,flag;

int AugPath[100];

int pathTrack;

char minCut[100][2];

int cutTrack;

int s[100];

int top;

void APclear(){

pathTrack=-1;

for(i=0;i<100;i++)

AugPath[i]=-1;

// printf("apc");

}

void APadd(int n){

AugPath[++pathTrack]=n;

// printf("apa");

}

int APempty(){

// printf("ape");

if(pathTrack==-1)

return 1;

else return 0;

}

void MCclear(){

// printf("mcc");

cutTrack=-1;

for(i=0;i<100;i++)

for(j=0;j<2;j++)

minCut[i][j]=-1;

}

void MCadd(int a,int b){

// printf("mca");

minCut[++cutTrack][0]=a;

minCut[cutTrack][1]=b;

}

void clear(){

// printf("c");

for(i=0;i<100;i++)

s[i]=-1;

top=-1;

}

int isEmpty(){

// printf("ie");

if(top==-1)

return 1;

return 0;

}

void push(int x){

// printf("push");

if(top<99)

s[++top]=x;

}

int pop(){

// printf("pop");

if(top>=0)

return s[top--];

}

int peek(){

// printf("peek");

if(top>=0 && top<100)

return s[top];

}

void Vclear(){

// printf("vc");

for(i=0;i<8;i++)

visited[i]=0;

}

void MaxFlow() {

while(findAugPath()!=0){

// printf("%d",pathTrack);

/\* for(i=0;i<=pathTrack;i++)

printf("%d",AugPath[i]); \*/

addFlow();

}

// printf(top);

Vclear();

clear();

do {

b=idSource();

}while(b!=0);

for(i=0;i<size;i++)

for(j=0;j<size;j++)

if(adj[i][j]!=0 && source[i]==1 && source[j]==0){

MCadd(i,j);

// printf("%d",i);

// printf("%d",j);

}

}

int findAugPath() {

APclear();

Vclear();

clear();

visited[0]=1;

push(0);

k=-1;

//printf("%d",isEmpty());

while(isEmpty()==0) {

k=getNextUnvNei(peek());

// printf("%d",k);

if(k==size-1) {

push(k);

break;

}

if(k==-1)

pop();

else {

visited[k]=1;

push(k);

}

}

if(k!=size-1)

return 0;

else{

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

APadd(s[i]);

// printf("%d",s[i]);

}

if(APempty()==1)

return 0;

else return 1;

}

}

int getNextUnvNei(int pos) {

// printf("gnun");

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

if(adj[pos][i]!=0 && visited[i]==0 && sat[pos][i]==0){

return i;

}

// printf("\_");

}

// printf("~");

return -1;

}

void addFlow() {

int min=adj[AugPath[0]][AugPath[1]];

for(i=1;i<pathTrack;i++) {

if(adj[AugPath[i]][AugPath[i+1]]<min)

min=adj[AugPath[i]][AugPath[i+1]];

}

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

flow[AugPath[i]][AugPath[i+1]]+=min;

if(flow[AugPath[i]][AugPath[i+1]]==adj[AugPath[i]][AugPath[i+1]])

sat[AugPath[i]][AugPath[i+1]]=1;

}

}

int idSource() {

clear();

visited[0]=1;

flag=0;

push(0);

source[0]=1;

while(isEmpty()==0) {

k=getNextUnvNei(peek());

if(k==-1)

pop();

else {

visited[k]=1;

source[k]=1;

push(k);

flag=1;

}

}

return flag;

}

void main(){

/\* FFMinimalCut g=new FFMinimalCut(8);

int[][] w={{0,40,60,0,0,0,0,0},{0,0,0,20,0,0,0,0},{0,0,0,10,15,0,0,0},{0,0,0,0,0,30,0,0},{0,0,0,0,0,0,10,0},{0,0,0,0,0,0,10,60},{0,0,0,0,0,0,0,50},{0,0,0,0,0,0,0,0}};

g.adj=w;

g.MaxFlow();

System.out.println("For g");

for(int i=0;i<g.minCut.size();i++)

System.out.println(g.minCut.get(i));

int[][] v= {{0,4,3,0},{0,0,1,3},{0,0,0,4},{0,0,0,0}};

FFMinimalCut h=new FFMinimalCut(4);

h.adj=v;

h.MaxFlow();

System.out.println("For h");

for(int i=0;i<h.minCut.size();i++)

System.out.println(h.minCut.get(i));

int[][] y= {{0,16,13,0,0,0},{0,0,10,12,0,0},{0,4,0,0,14,0},{0,0,9,0,0,20},{0,0,0,7,0,4},{0,0,0,0,0,0}};

FFMinimalCut x=new FFMinimalCut(6);

x.adj=y;

x.MaxFlow();

System.out.println("For x");

for(int i=0;i<x.minCut.size();i++)

System.out.println(x.minCut.get(i)); \*/

int data[MAXROW][MAXCOL];

//char a[40];

int flag=0;

int i, j;

int r, c;

char s1[80],s2[80];

unsigned char ch;

FILE \*f;

f=fopen("wf5.pgm", "r");

if (f==NULL)

{

printf("Cannot open");

exit(1);

}

fgets(s1,80,f);

while(atoi(s1)!=255)

{

strcpy(s2,s1);

fgets(s1,80,f);

}

i=0;

while(s2[i]!=' ')

{

s1[i]=s2[i];

i++;

}

s1[i]='\0';

c=atoi(s1);

j=0;

while(s2[i]==' ')

i++;

while(s2[i]!='\0')

{

s1[j]=s2[i];

j++;

i++;

}

r=atoi(s1);

printf("Col:%d, Row:%d\n",c,r);

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

ch=fgetc(f);

data[i][j]=ch;

}

}

fclose(f);

f=fopen("testout.pgm", "w");

fprintf(f, "P2\n");

fprintf(f, "%d %d\n", c, r);

fprintf(f, "255\n");

MaxFlow();

for (i=0; i<r; i++)

{

for (j=0; j<c; j++)

fprintf(f, "%d\t", data[i][j]);

fprintf(f, "\n");

}

/\* for (i=0; i<r; i++)

{

for (j=0; j<c; j++)

printf("%d\t", data[i][j]);

}\*/

fclose(f);

/\* for(i=0;i<=cutTrack;i++){

printf("%d",minCut[i][0]);

printf("%d",minCut[i][1]);

} \*/

}