НТУУ “КПІ”

Кафедра Обчислювальної техніки

**Лабораторна робота №6**

**з дискретної математики**

Виконав

ст. І курсу

ФІОТ, гр. ІО-82

Куцовол Віктор

Київ 2009

**Варіант №13**

Граф С1:



Матриця суміжності:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **V1** | **V2** | **V3** | **V4** | **V5** | **V6** | **V7** | **V8** | **V9** | **V10** | **V11** |
| **V1** | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| **V2** | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **V3** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| **V4** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| **V5** | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| **V6** | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| **V7** | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| **V8** | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| **V9** | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| **V10** | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **V11** | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Графі С2



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **V1** | **V2** | **V3** | **V4** | **V5** | **V6** | **V7** | **V8** | **V9** |
| **V1** | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| **V2** | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| **V3** | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| **V4** | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| **V5** | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| **V6** | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| **V7** | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| **V8** | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| **V9** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

У вихідному файлі програми бачимо, що існує 85 ланцюгів з вершини **1** у вершину **9**. Це:

1 2 3 4 5 8 9

1 2 3 4 5 9

1 2 3 4 6 7 5 8 9

1 2 3 4 6 7 5 9

1 2 3 4 6 7 8 5 9

1 2 3 4 6 7 8 9

1 2 3 8 5 9

1 2 3 8 7 5 9

1 2 3 8 7 6 4 5 9

1 2 3 8 9

1 2 5 4 3 8 9

1 2 5 4 6 7 8 9

1 2 5 8 9

1 2 5 9

1 2 7 5 4 3 8 9

1 2 7 5 8 9

1 2 7 5 9

1 2 7 6 4 3 8 5 9

1 2 7 6 4 3 8 9

1 2 7 6 4 5 8 9

1 2 7 6 4 5 9

1 2 7 8 3 4 5 9

1 2 7 8 5 9

1 2 7 8 9

1 5 2 3 4 6 7 8 9

1 5 2 3 8 9

1 5 2 7 6 4 3 8 9

1 5 2 7 8 9

1 5 4 3 2 7 8 9

1 5 4 3 8 9

1 5 4 6 7 2 3 8 9

1 5 4 6 7 8 9

1 5 8 9

1 5 9

1 6 4 3 2 5 8 9

1 6 4 3 2 5 9

1 6 4 3 2 7 5 8 9

1 6 4 3 2 7 5 9

1 6 4 3 2 7 8 5 9

1 6 4 3 2 7 8 9

1 6 4 3 8 5 9

1 6 4 3 8 7 2 5 9

1 6 4 3 8 7 5 9

1 6 4 3 8 9

1 6 4 5 2 3 8 9

1 6 4 5 2 7 8 9

1 6 4 5 8 9

1 6 4 5 9

1 6 7 2 3 4 5 8 9

1 6 7 2 3 4 5 9

1 6 7 2 3 8 5 9

1 6 7 2 3 8 9

1 6 7 2 5 4 3 8 9

1 6 7 2 5 8 9

1 6 7 2 5 9

1 6 7 5 2 3 8 9

1 6 7 5 4 3 8 9

1 6 7 5 8 9

1 6 7 5 9

1 6 7 8 3 2 5 9

1 6 7 8 3 4 5 9

1 6 7 8 5 9

1 6 7 8 9

1 7 2 3 4 5 8 9

1 7 2 3 4 5 9

1 7 2 3 8 5 9

1 7 2 3 8 9

1 7 2 5 4 3 8 9

1 7 2 5 8 9

1 7 2 5 9

1 7 5 2 3 8 9

1 7 5 4 3 8 9

1 7 5 8 9

1 7 5 9

1 7 6 4 3 2 5 8 9

1 7 6 4 3 2 5 9

1 7 6 4 3 8 5 9

1 7 6 4 3 8 9

1 7 6 4 5 2 3 8 9

1 7 6 4 5 8 9

1 7 6 4 5 9

1 7 8 3 2 5 9

1 7 8 3 4 5 9

1 7 8 5 9

1 7 8 9

Код програми:

Модуль:

unit chain;

interface

const

size=20;

type

TIndex=1..size;

TRow=array[TIndex] of byte;

TMat=array[TIndex] of TRow;

procedure InputMat(var n:TIndex;var A:TMat);

procedure FindWay(var n:TIndex;var A:TMat;f,l:TIndex);

implementation

procedure InputMat(var n:TIndex;var A:TMat);

var

i,k:TIndex;

begin

writeln('Input size');

readln(n);

writeln('Input matrix');

for i:=1 to n do

begin

for k:=1 to n do

read(a[i,k]);

readln

end;

end;

procedure FindWay(var n:TIndex;var A:TMat;f,l:TIndex);

var

i,k,j:integer;

b,c:TRow;

begin

if f<>l then

begin

for i:=1 to n do

begin

b[i]:=0;

c[i]:=0;

end;

b[1]:=f;

c[f]:=1;

i:=1;

k:=1;

while i>=1 do

begin

if (c[k]=0) and (a[b[i],k]=1) then

begin

c[k]:=1;

i:=i+1;

b[i]:=k;

k:=0;

end;

if b[i]=l then

begin

for j:=1 to i do write(b[j],' ');

writeln();

end;

k:=k+1;

while (k>n) or (b[i]=l) do

begin

k:=b[i]+1;

c[k-1]:=0;

i:=i-1;

end;

end;

end;

end;

end.

Програма:

program lab5;

uses chain;

var

a:TMat;

n,f,l:TIndex;

begin

InputMat(n,a);

writeln('Input first');

readln(f);

writeln('Input last');

readln(l);

FindWay(n,a,f,l);

readln

end.