МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №5

з дисципліни «Дискретна математика»

на тему: *«Розфарбовування графа, алгоритми розфарбування»*

Виконав:

студент гр.ІО-31

Долинний О.В.

Перевірив:

Новотарський М.А.

Київ 2014

**Лабораторна робота №5**

**Тема:** «Розфарбовування графа, алгоритми розфарбування».

**Мета:** вивчення способів правильного розфарбовування графа.

**Завдання:** створити програму для правильного розфарбовування графа на

основі одного з алгоритмів розфарбування.

1. ***Завдання до лабораторної роботи***

Необхідно в рамках форми OperForm реалізувати алгоритм відповідно до варіанту лабораторної роботи.

Номер варіанту I визначається як результат операції I = NZK mod 6+1, де NZK – номер залікової книжки. При виконанні завдань лабораторної роботи використати такі варіанти задавання графа:

**І= 3110 mod 6 + 1 = 3**

***Варіант 3:***

А) Виконати завдання 3 до лабораторної роботи.

Б) Программа повинна дозволяти розфарбування довільного.

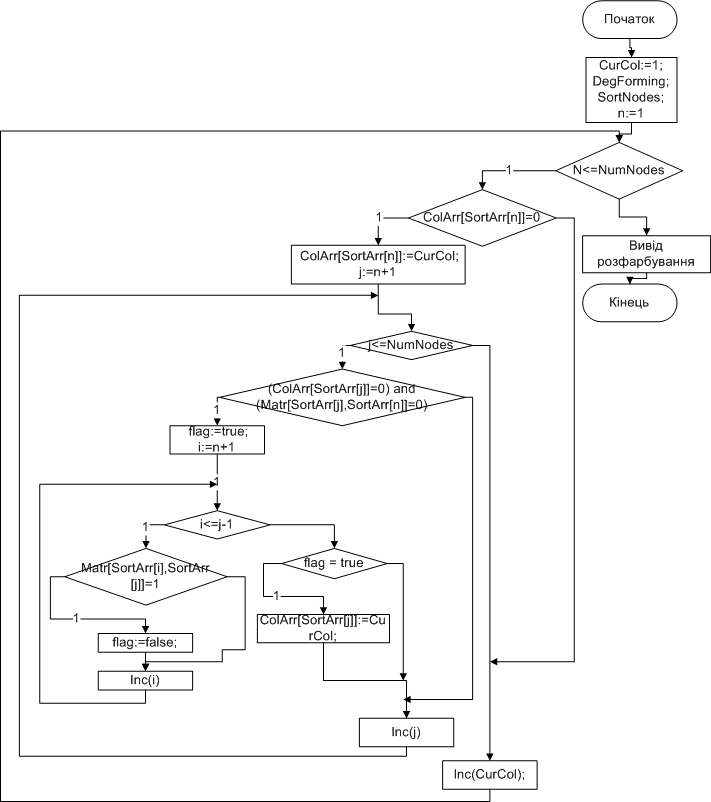
В) Перевірити роботу програми на даному графі G. Вивести у графічному режимі розфарбований граф, або включити у протокол розфарбований вручну граф за результатами роботи програми.

******

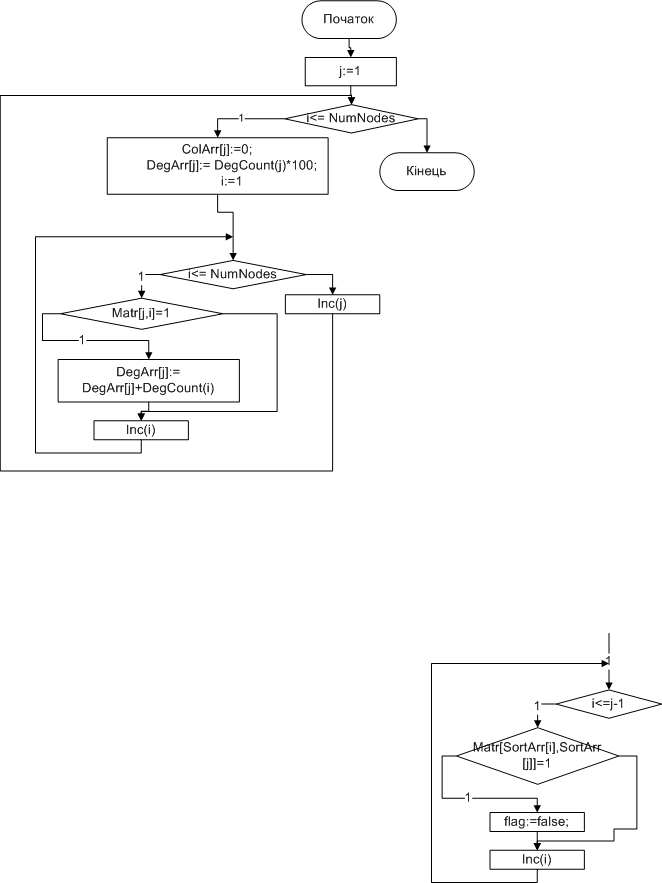
***Завдання 3:*** Вивчити основні означення та теореми про розфарбування графів. Створити програму розфарбування графів яка реалізує модифікований евристичний алгоритм розфарбування.

1. ***Блок-схема алгоритму***

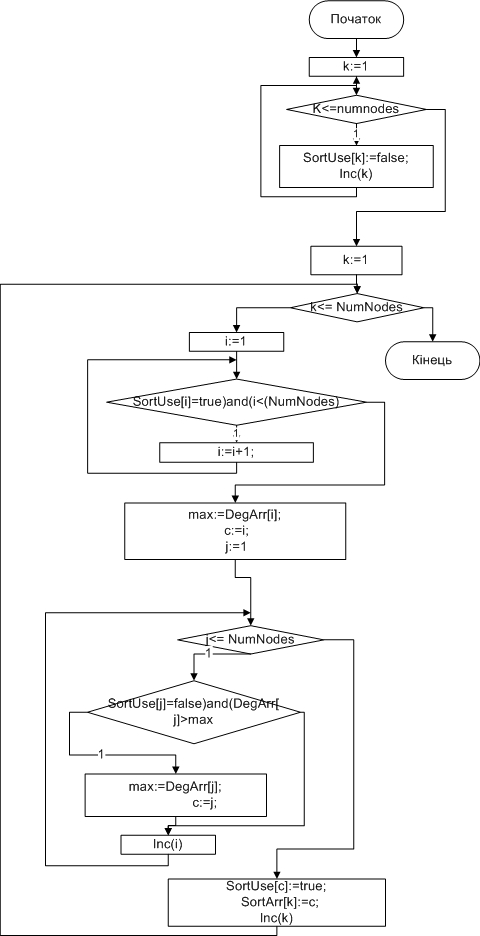
**Процедура PaintingClick**

****

**Процедура DegForming**

****

**Процедура SortNodes**

****

1. ***Роздруківка тексту програми***

Const Nmax=100; {\*максимальна кількість вершин графа\*}

type

TArr = Array [1..Nmax] of Integer;

TA = Array [1..Nmax, 1..Nmax] of Byte;

TBoolean = Array[1..Nmax] of boolean;

var

OperForm: TOperForm;

Matr: TA; {\*матриця суміжності графа\*}

ColArr: TArr; {\*масив номерів фарб для кожної вершини графа\*}

DegArr: TArr; {\*масив ступенів вершин\*}

SortArr:TArr;{\*відсортований за зменшенням ступенів масив вершин\*}

CurCol: Byte; {\*поточний номер фарби\*}

SortUse:TBoolean;

procedure TOperForm.PaintGraph; // отрисовка графа

var i,j:Integer;

h,x,y,x0,y0,x1,y1:Integer;

Angle,a:Extended;

begin

GrImage.Canvas.Pen.Width:= 1; // Толщина окаймляющей линии

GrImage.Canvas.Pen.Color:=clBlack; // Цвет окаймляющей линии

GrImage.Canvas.Brush.Color := clwhite; // Цвет фона

GrImage.Canvas.Rectangle(0, 0, 600, 600);// Размер картинки

GrImage.Canvas.Font.Size:=10;

GrImage.Canvas.Pen.Color:=clBlack; // Цвет ребер

X0:=300;

Y0:=300;

a:=0; // вершины ставим начиная с 1, против часовой стрелки

h:=1; // номер вершины

GrImage.Canvas.Pen.Width:= 2;

GrImage.Canvas.Pen.Color:=clBlack;

While a<360 do

begin

CoordArr[h].X:=x0+Round( 220 \* cos(a\*2\*pi/360)); //массив координат

CoordArr[h].Y:=x0-Round( 220 \* sin(a\*2\*pi/360));

GrImage.Canvas.MoveTo(CoordArr[h].X,CoordArr[h].Y);

GrImage.Canvas.Ellipse(CoordArr[h].X-6,CoordArr[h].Y-6,CoordArr[h].X+6,CoordArr[h].Y+6);

// цифры по большему радиусу

If (a>45 ) AND (a<225) then

begin

x:=x0+Round( (220+30) \* cos(a\*2\*pi/360));

y:=x0-Round( (220+30) \* sin(a\*2\*pi/360));

end

else

begin

x:=x0+Round( (220+15) \* cos(a\*2\*pi/360));

y:=x0-Round( (220+15) \* sin(a\*2\*pi/360));

end;

GrImage.Canvas.TextOut(x-2,y-2,IntToStr(h));

Inc(h);

a:=a+360/NumNodes;

end;

For i:=1 to NumNodes do

For j:=1 to NumNodes do

begin

If ResultGrid.Cells[j,i]='1' then

begin

If i<>j then

begin // Отрисовка ребер, соединяющих различные вершины

GrImage.Canvas.MoveTo(CoordArr[i].X,CoordArr[i].Y);

GrImage.Canvas.LineTo(CoordArr[j].X,CoordArr[j].Y);

end else

begin // Отрисовка петель

Angle:=(i-1)\*360/NumNodes;

x1:=CoordArr[i].X+Round( 25 \* cos(Angle\*2\*pi/360));

y1:=CoordArr[i].Y-Round( 25 \* sin(Angle\*2\*pi/360));

GrImage.Canvas.Ellipse(x1-25,y1-25,x1+25,y1+25);

end;

end;

end;

end;

procedure TOperForm.DownloadClick(Sender: TObject);

var F : textfile;

Str:String;

m,i,j,x,y,x0,y0,x1,y1,h:Integer;

Angle,a:Extended;

begin

AssignFile(F,'DATA\P3.TXT');

Reset(F);

Readln(F, Str);

ResultGrid.RowCount:=StrToInt(str);

Readln(F, Str);

ResultGrid.ColCount:=StrToInt(str);

NumNodes:=ResultGrid.ColCount-1;

For i:=1 to ResultGrid.RowCount-1 do

For j:=1 to ResultGrid.ColCount-1 do

begin

Readln(F,str);

if(str = '1') then

ResultGrid.Cells[j,i]:='1'

else

ResultGrid.Cells[j,i]:='0';

end;

CloseFile(F);

For i:=1 to ResultGrid.RowCount-1 do

For j:=1 to ResultGrid.ColCount-1 do

begin

if (ResultGrid.Cells[j,i]='0') and (ResultGrid.Cells[i,j]='1') then

ResultGrid.Cells[j,i]:='1';

end;

For i:=1 to NumNodes do

begin

ResultGrid.Cells[i,0]:=IntToStr(i);

ResultGrid.Cells[0,i]:=IntToStr(i);

end;

For i:=1 to ResultGrid.RowCount-1 do

For j:=1 to ResultGrid.ColCount-1 do

Matr[j,i]:=strtoint(ResultGrid.Cells[i,j]);

PaintGraph;

end;

procedure TOperForm.PaintingClick(Sender: TObject);

var i,j,n : integer;

flag:boolean;

begin

CurCol:=1;

DegForming; {\*Формування масиву ступенів вершин\*}

SortNodes; {\*Формування массиву відсортованих вершин SortArr\*}

For n:=1 to NumNodes do

begin

If ColArr[SortArr[n]]=0 then

begin

ColArr[SortArr[n]]:=CurCol;

For j:=n+1 to NumNodes do

begin

if (ColArr[SortArr[j]]=0) and (Matr[SortArr[j],SortArr[n]]=0) then

begin

flag:=true;

for i:=n+1 to j do

if (Matr[SortArr[i],SortArr[j]]=1) and (ColArr[SortArr[i]]=CurCol)then flag:=false;

if flag = true then ColArr[SortArr[j]]:=CurCol;

end;

end;

Inc(CurCol);

end;

end;

{\*<Виводимо результат розфарбування>\*}

For i:=1 to NumNodes do

ColloringTopOf(i,ColArr[i]);

For i:=1 to NumNodes do

SortRes.Cells[i,0]:=IntToStr(SortArr[i]);

SortRes.Cells[0,0]:=IntToStr(numnodes);

end;

Procedure TOperForm.DegForming; {\*Процедура формування масиву ступенів вершин\*}

Var i,j,n:Byte;

Function DegCount(m:Byte):Integer;

Var k, Deg:Integer;

Begin

Deg:=0;

For k:=1 to NumNodes do Deg:= Deg+Matr[k,m];

DegCount:=Deg;

End;

Begin

For j:=1 to NumNodes do

begin

ColArr[j]:=0;

DegArr[j]:= DegCount(j)\*100;

For i:=1 to NumNodes do

If (Matr[j,i]=1 )then DegArr[j]:= DegArr[j]+DegCount(i);

end;

End;

Procedure TOperForm.SortNodes; {\*Сортування вершин за ступенями\*}

Var c,k,i,j:Byte; max:integer;

Begin

for k:=1 to numnodes do

SortUse[k]:=false;

For k:=1 to NumNodes do

begin

i:=1;

while ( SortUse[i]=true)and(i<(NumNodes)) do

i:=i+1;

max:=DegArr[i];

c:=i;

for j:=i to NumNodes do

if (SortUse[j]=false)and(DegArr[j]>max) then

begin

max:=DegArr[j];

c:=j;

end;

SortUse[c]:=true;

SortArr[k]:=c;

end;

End;

procedure TOperForm.ColloringTopOf(number: integer; colorOf: integer);

var i:Integer;

x,y: integer; // координаты номеров вершин

x0,y0: integer; // центр графа

a: Extended; // угол между OX и прямой (x0,y0) (x,y)

h: integer; // номера вершины

begin

//выбор цвета, в который будут окрашена вершина с номером i

case colorOf of

1: GrImage.Canvas.Pen.Color:=clRed;

2: GrImage.Canvas.Pen.Color:=clBlue;

3: GrImage.Canvas.Pen.Color:=clLime;

4: GrImage.Canvas.Pen.Color:=clYellow;

5: GrImage.Canvas.Pen.Color:=clAqua;

6: GrImage.Canvas.Pen.Color:=clOlive;

7: GrImage.Canvas.Pen.Color:=clTeal;

8: GrImage.Canvas.Pen.Color:=clGreen;

9: GrImage.Canvas.Pen.Color:=clMaroon;

10: GrImage.Canvas.Pen.Color:=clFuchsia;

11: GrImage.Canvas.Pen.Color:=clGray;

end;

X0:=300;

Y0:=300;

a:=0; // вершины ставим начиная с 1, против часовой стрелки

h:=1; // номер вершины

GrImage.Canvas.Pen.Width:= 9;

While a<360 do

begin

CoordArr[h].X:=x0+Round( 220 \* cos(a\*2\*pi/360)); //массив координат

CoordArr[h].Y:=x0-Round( 220 \* sin(a\*2\*pi/360));

if number = h then

begin

GrImage.Canvas.MoveTo(CoordArr[h].X,CoordArr[h].Y);

GrImage.Canvas.Ellipse(CoordArr[h].X-1,CoordArr[h].

Y-1,CoordArr[h].X+1,CoordArr[h].Y+1);

end;

// цифры по большему радиусу

If (a>45 ) AND (a<225) then

begin

x:=x0+Round( (220+30) \* cos(a\*2\*pi/360));

y:=x0-Round( (220+30) \* sin(a\*2\*pi/360));

end else

begin

x:=x0+Round( (220+15) \* cos(a\*2\*pi/360));

y:=x0-Round( (220+15) \* sin(a\*2\*pi/360));

end;

GrImage.Canvas.TextOut(x-2,y-2,IntToStr(h));

// относит. степень

If (a>45 ) AND (a<225) then

begin

x:=x0+Round( (240+30) \* cos(a\*2\*pi/360));

y:=x0-Round( (240+30) \* sin(a\*2\*pi/360));

end else

begin

x:=x0+Round( (240+15) \* cos(a\*2\*pi/360));

y:=x0-Round( (240+15) \* sin(a\*2\*pi/360));

end;

GrImage.Canvas.TextOut(x-2,y-2,IntToStr(DegArr[h] div 100));

// двойная относит степень

If (a>45 ) AND (a<225) then

begin

x:=x0+Round( (260+30) \* cos(a\*2\*pi/360));

y:=x0-Round( (260+30) \* sin(a\*2\*pi/360));

end else

begin

x:=x0+Round( (260+15) \* cos(a\*2\*pi/360));

y:=x0-Round( (260+15) \* sin(a\*2\*pi/360));

end;

GrImage.Canvas.TextOut(x-2,y-2,IntToStr(DegArr[h] mod 100));

Inc(h);

a:=a+360/NumNodes;

end;

end;

1. ***Контрольний приклад та результати машинного розрахунку***

Розфарбувати граф, вказаний на рисунку:

******

Відсортуємо вершини графа за зменшенням їх ступенів. У результаті отримуємо вектор відсортованих вершин SortArr = (7,2,4,5,6,8,9,10,1,3)

Вектор ступенів відсортованих вершин має наступний вигляд: D = (7,5,4,4,4,4,3,3,3,3)

У першому рядку таблиці запишемо вектор SortArr, у другому – вектор D , а у третьому – вектор D2 = (27, 21,19,19,17,17,14,14,13,13).

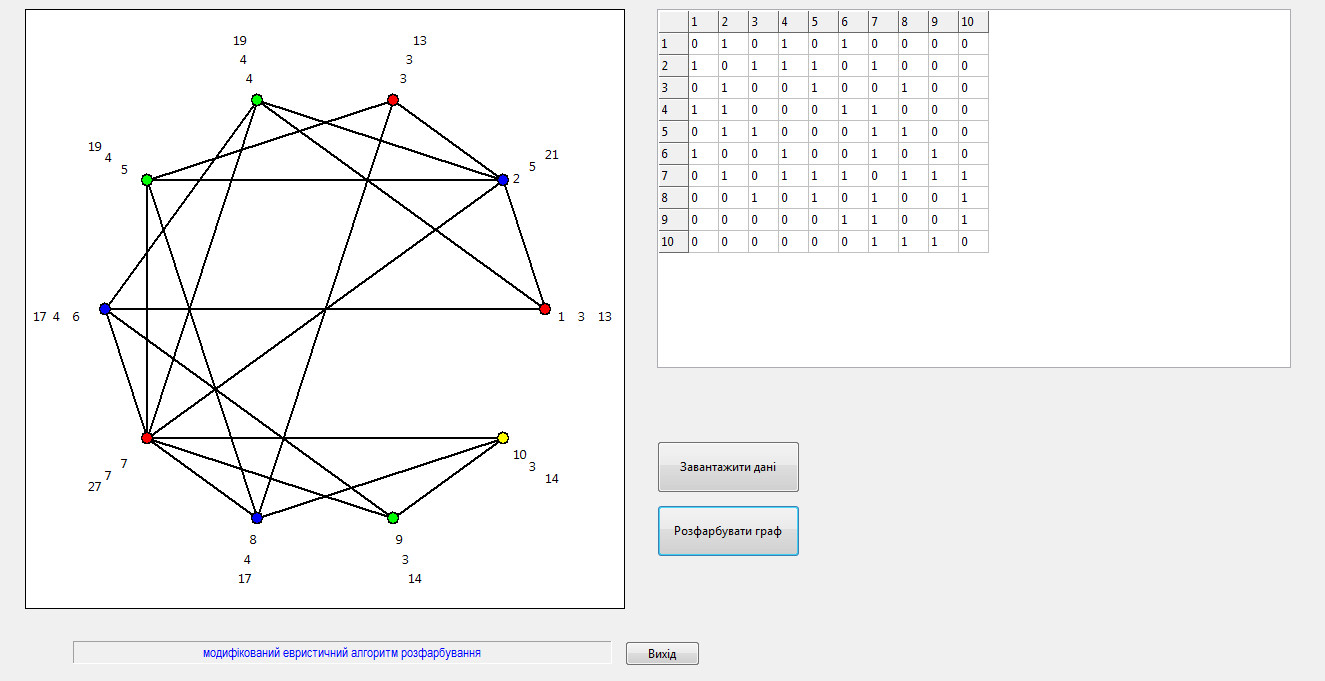
Четвертий рядок відповідає представленню ступенів D та D2 в масиві DegArr.

Наступні рядки відображають вміст вектора розфарбування col(X ).

Таким чином, даний граф можна розфарбувати не менш ніж у чотири кольори, тобто γ (G) = 4 .

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номери вершин | а7 | а2 | а4 | а5 | а6 | а8 | а9 | а10 | а1 | а3 |
| Ступінь вершин D | 7 | 5 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 |
| D2 | 27 | 21 | 19 | 19 | 17 | 17 | 14 | 14 | 13 | 13 |
| DegArr | 727 | 521 | 419 | 419 | 417 | 417 | 314 | 314 | 313 | 313 |
| CurCol=1 | 1 | - | - | - | - | - | - | - | 1 | 1 |
| CurCol=2 | 1 | 2 | - | - | 2 | 2 | - | - | 1 | 1 |
| CurCol=3 | 1 | 2 | 3 | 3 | 2 | 2 | 3 | - | 1 | 1 |
| CurCol=4 | 1 | 2 | 3 | 3 | 2 | 2 | 3 | 4 | 1 | 1 |

******



1. ***Висновки по роботі***

*У ході лабораторної роботи я закріпив знання з теорії графів, навчився використовувати набуті знання на практиці. При виконанні роботи граф було розфарбовано за допомогою модифікованого евристичного методу.*

*Отримані результати підтверджують правильність роботи алгоритму.*

*Під час виконання лабораторної роботи я покращив навики роботи за програмами з візуальними частинами.*