

НТУУ «КП»

ФІОТ

Кафедра ОТ

## Лабораторна робота №5

Виконав:

студент групи ІО-34

Власов Максим

Номер залікової книжки:

3405

Київ 2014р.

**Тема:** «Розфарбовування графа, алгоритми розфарбовування»

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

**Завдання:** створити програму для правильного розфарбовування графа на основі одного з алгоритмів розфарбовування.

$$I = 3405 \bmod 6 + 1 = 3$$

### Теорія:

Різноманітні завдання, що виникають при плануванні виробництва, складанні графіків огляду, зберіганні та транспортуванні товарів та ін., часто можуть бути представлені як задачі теорії графів, тісно пов'язані з так званим «завданням розфарбовування». Графи, що розглядаються в даній лабораторній роботі, є неорієнтованими і такими, що не мають петель.

Граф  $G$  називають  $r$ -хроматичним, якщо його вершини можуть бути розфарбовані з використанням  $r$  кольорів (фарб) так, що не знайдеться двох суміжних вершин одного кольору. Найменше число  $r$ , таке, що граф  $G$  є  $r$ -хроматичним, називається хроматичним числом графа  $G$  і позначається  $\gamma(G)$ . Завдання знаходження хроматичного числа графа називається задачею про розфарбовування (або завданням розфарбовування) графа. Відповідне цьому числу розфарбування вершин розбиває множину вершин графа на  $r$  підмножин, кожна з яких містить вершини одного кольору. Ці множини є незалежними, оскільки в межах однієї множини немає двох суміжних вершин.

Завдання знаходження хроматичного числа довільного графа стало предметом багатьох досліджень в кінці XIX і в XX столітті. З цього питання отримано багато цікавих результатів.

Хроматичне число графа не можна знайти, знаючи тільки кількість вершин і ребер графа. Недостатньо також знати степінь кожної вершини, щоб обчислити хроматичне число графа. При відомих величинах  $n$  (кількість вершин),  $m$  (кількість ребер) і  $\deg(x_1), \dots, \deg(x_n)$  (степені вершин графа) можна отримати тільки верхню і нижню оцінки для хроматичного числа графа.

-  $x_1$                       -  $x_2$                       -  $x_3$

*Теорема про п'ять фарб.* Кожен планарний граф можна розфарбувати за допомогою п'яти кольорів так, що будь-які дві суміжні вершини будуть пофарбовані в різні кольори, тобто якщо граф  $G$  - планарний, то  $\gamma(G) \leq 5$ .

*Гіпотеза про чотири фарби (недоведена).* Кожен планарний граф можна розфарбувати за допомогою чотирьох кольорів так, що будь-які дві суміжні вершини будуть пофарбовані в різні кольори, тобто якщо граф  $G$  – планарний, то  $\gamma(G) \leq 4$ .

У 1852 р. про гіпотезу чотирьох фарб говорилося в листуванні Огюста де Моргана з сером Вільямом Гамільтоном. З того часу ця «теорема» стала, поряд з теоремою Ферма, однією з найзнаменитіших невирішених задач в математиці.

*Повний граф  $K_n$*  завжди розфарбовується в  $n$  кольорів, тобто кількість кольорів дорівнює кількості його вершин.

Приклад розфарбовування графа наведено на рисунку 5.1. Цей граф є однією із заборонених фігур, що використовуються для визначення планарності. Цифрами «1» і «2» позначені кольори вершин.

Максимальна кількість незалежних вершин графа  $\alpha(G)$ , що дорівнює потужності найбільшої множини попарно несуміжних вершин, збігається також з потужністю найбільшої множини вершин в  $G$ , які можуть бути пофарбовані в один колір, отже:

$$\gamma(G) \geq \left\lceil \frac{n}{\alpha(G)} \right\rceil, \quad (5.1)$$

де  $n$  - кількість вершин графа  $G$ , а  $\lceil x \rceil$  позначає найбільше ціле число, яке не більше за  $x$ .

Ще одна нижня оцінка для  $\gamma(G)$  може бути отримана наступним чином:

$$\gamma(G) \geq \frac{n^2}{n^2 - 2m}. \quad (5.2)$$

Верхня оцінка хроматичного числа може бути обчислена за формулою:

$$\gamma(G) \leq 1 + \max_{x_j \in X} [d(x_j) + 1]. \quad (5.3)$$

Застосування оцінок для хроматичного числа значно звужує межі рішення. Для визначення оцінки хроматичного числа також можуть використовуватися інші топологічні характеристики графа, наприклад, властивість планарності.

Граф, який можна зобразити на площині так, що жодні два його ребра не перетинаються між собою, називається *планарним*.

## **Код програми:**

```
unit Unit2;
```

```
{ $mode objfpc } { $H+ }
```

```
interface
```

```
uses
```

```
Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, ExtCtrls,  
StdCtrls, Grids, Unit3, Unit5;
```

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

```
{ TOperForm }
```

```
TOperForm = class(TForm)
```

```

Download: TButton;
Painting: TButton;
ExitButton: TButton;
GrImage: TImage;
InfoPanel: TPanel;
ResultGrid: TStringGrid;
procedure DownloadClick(Sender: TObject);
procedure PaintingClick(Sender: TObject);
procedure ExitButtonClick(Sender: TObject);
procedure ColloringTopOf(number: integer; colorOf: integer);
procedure PaintGraph;
procedure DegForming;
procedure SortNodes;
private
  { private declarations }
public
  { public declarations }
end;

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;  { *м а т р и ц я с у м і ж н о с т і г р а ф а * }
  ColArr: TArr; { *м а с и в н о м е р і в ф а р б д л я к о ж н о ї в е р ш и н и г р а
ф а * }
  DegArr: TArr; { *м а с и в с т у п е н і в в е р ш и н * }
  SortArr: TArr; { *в і д с о р т о в а н и й з а з м е н ш е н н я м с т у п е н і в м а с
и в в е р ш и н * }
  CurCol: Byte; { *п о т о ч н и й н о м е р ф а р б и * }
  SortUse: TBoolean;

implementation

{$R *.Ifm}

{ TOperForm }

procedure TOperForm.ExitButtonClick(Sender: TObject);
begin
  Close;
end;

procedure TOperForm.PaintGraph;
var i,j: Integer;
    h,x,y,x0,y0,x1,y1: Integer;
    Angle,a: Extended;

```

```

begin
  GrImage.Canvas.Pen.Width:=1;
  GrImage.Canvas.Pen.Color:=clBlack
  GrImage.Canvas.Brush.Color:=clwhite;
  GrImage.Canvas.Rectangle(0, 0, 600, 600);
  GrImage.Canvas.Font.Size:=10;
  GrImage.Canvas.Pen.Color:=clBlack;
  X0:=300;
  Y0:=300;
  a:=0;
  h:=1;
  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));
      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);

      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
                GrImage.Canvas.MoveTo(CoordArr[i].X,CoordArr[i].Y);
                GrImage.Canvas.LineTo(CoordArr[j].X,CoordArr[j].Y);
              end else
                begin
                  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 TForm1.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;
        end;
    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;
        end;
    end;
    For i:=1 to NumNodes do
        begin
            ResultGrid.Cells[i,0]:=IntToStr(i);
            ResultGrid.Cells[0,i]:=IntToStr(i);
        end;
    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]);
        end;
    end;
    PaintGraph;
end;

```

```

procedure TForm1.PaintingClick(Sender: TObject);
var i,j,n : integer;
    flag:boolean;

```

```

begin
  CurCol:=1;
  DegForming;{*Ф о р м у в а н н я м а с и в у с т у п е н і в в е р ш и н *}
  SortNodes;{*Ф о р м у в а н н я м а с с и в у в і д с о р т о в а н и х в е р ш и н
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;
            end;
          Inc(CurCol);
        end;
      end;
    end;
  {*<В и в о д и м о р е з у л ь т а т р о з ф а р б у в а н н я> *}
  For i:=1 to NumNodes do
    ColloringTopOf(i,ColArr[i]);
  end;

```

```

Procedure TOperForm.DegForming;{*П р о ц е д у р а ф о р м у в а н н я м а с и в у с т
у п е н і в в е р ш и н *}
  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;
    end;
  end;

```

End;

Procedure TOperForm.SortNodes;{\*С о р т у в а н н я в е р ш и н з а с т у п е н я м и\*}

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;

x0,y0: integer;

a: Extended;

h: integer;

begin

case colorOf of

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

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

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

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

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

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

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

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

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

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

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

end;

X0:=300;

Y0:=300;

a:=0;

h:=1;



```

GrImage.Canvas.Pen.Width:=9;
While a<360 do
begin
CoordArr[h].X:=x0+Round( 220 * cos(a*2*pi/360));
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;

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));

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));

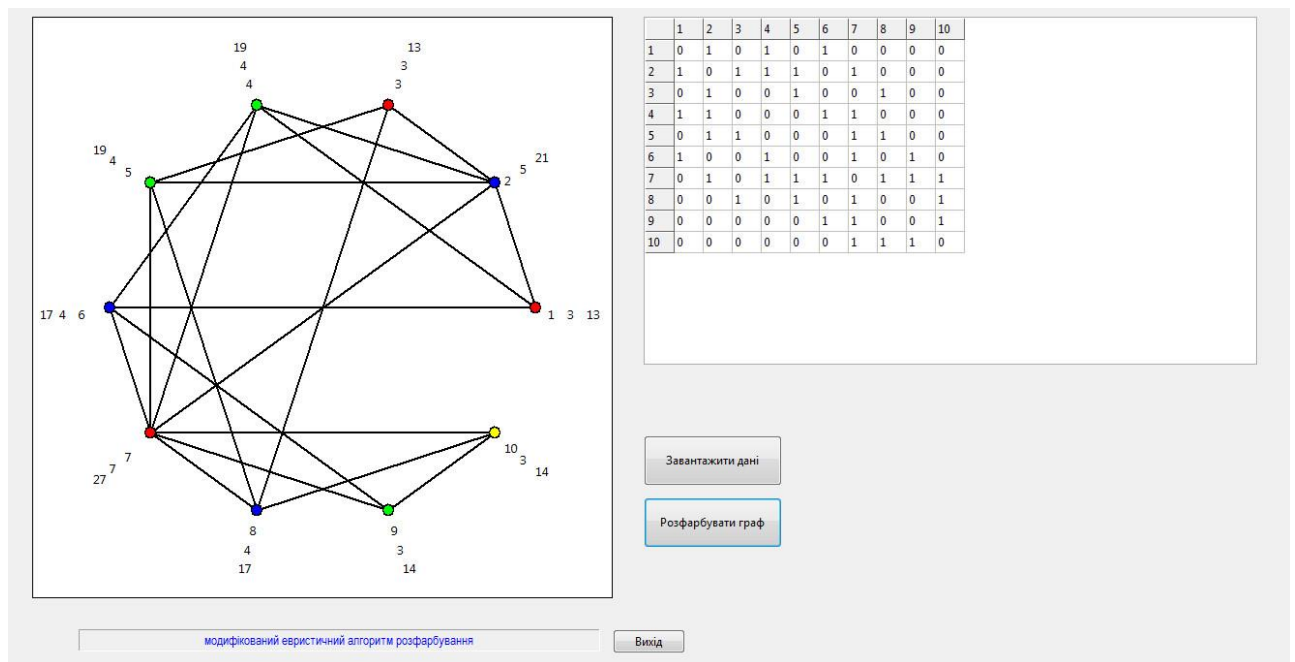
If (a>45 ) AND (a<225) then
begin
x:=x0+Round( (260+30) * cos(a*2*pi/360));
y:=x0-Round( (260+30) * sin(a*2*pi/360));
end else
begin
x:=x0+Round( (260+15) * cos(a*2*pi/360));
y:=x0-Round( (260+15) * sin(a*2*pi/360));
end;
GrImage.Canvas.TextOut(x-2,y-2,IntToStr(DegArr[h] mod 100));

Inc(h);
a:=a+360/NumNodes;
end;
end;

```

end.

## Скріншоти:



## Висновки:

При виконанні роботи граф було розфарбовано за допомогою модифікованого евристичного методу.

Отримані результати підтверджують правильність роботи алгоритму.