

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ, УПРАВЛЕНИЯ И ДИЗАЙНА
"КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ"**

**Лабораторная работа №5
по курсу "Алгоритмы и Структуры данных"
на тему: "Представления графов в ЭВМ"
Вариант № 13**

**Выполнил студент группы ВТ-21:
Потеренко А.Г.
Проверил преподаватель:
Мингалиев Р.Ш.**

Порядок работы.

1. Анализ индивидуального задания и разработка способов представления объектов задачи в памяти, методов доступа к ним.
2. Разработка программы на языке Паскаль.
3. Разработка контрольных примеров.
4. Отладка программ.
5. Составление отчета.

Содержание отчета.

	Стр.
1. Текст постановки задачи.....	3
2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.....	3
3. Алгоритм.....	4
4. Описание и обоснование контрольных примеров.....	5-6
5. Текст программы с комментариями (в виде приложения).....	6-19

1. Текст постановки задачи.

Необходимо построить граф указанным способом представления, построить остовое дерево, используя поиск в глубину и в ширину, затем перевести граф во вторую форму представления и повторить алгоритмы, использованные для первой формы. Оценить скорость работы алгоритмов на первой и второй формах представления. Построить итоговую таблицу вида:

Размерность (количество вершин)	Первый способ представления		Второй способ представления	
	Поиск в глубину	Поиск в ширину	Поиск в глубину	Поиск в ширину
2				
3				
...				
50				

Вариант задания:

Способы представления графов:

- 1 Матрица смежности.
- 5 Одномерный массив.

2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.

Так как граф представлен матрицей смежности, то мы создаем матрицу:

$M[1..n, 1..n]$. Каждый элемент поля представлен двумя значениями 0 и 1. Если значение 0 – то данная вершина строки не имеет смежности с вершиной столбца, если 1 – то имеет смежность.

Одномерный массив представлен таким образом: первые n элементов равны указателям на данные вершины. Указатель указывает на поле, в котором указывается число смежных вершин у данной вершины. После этого поля следует перечисление каждой смежной вершины. Одномерный массив имеет порядок $1..n*n+n$.

3. Алгоритм решения данной задачи.

1.Обход в глубину.

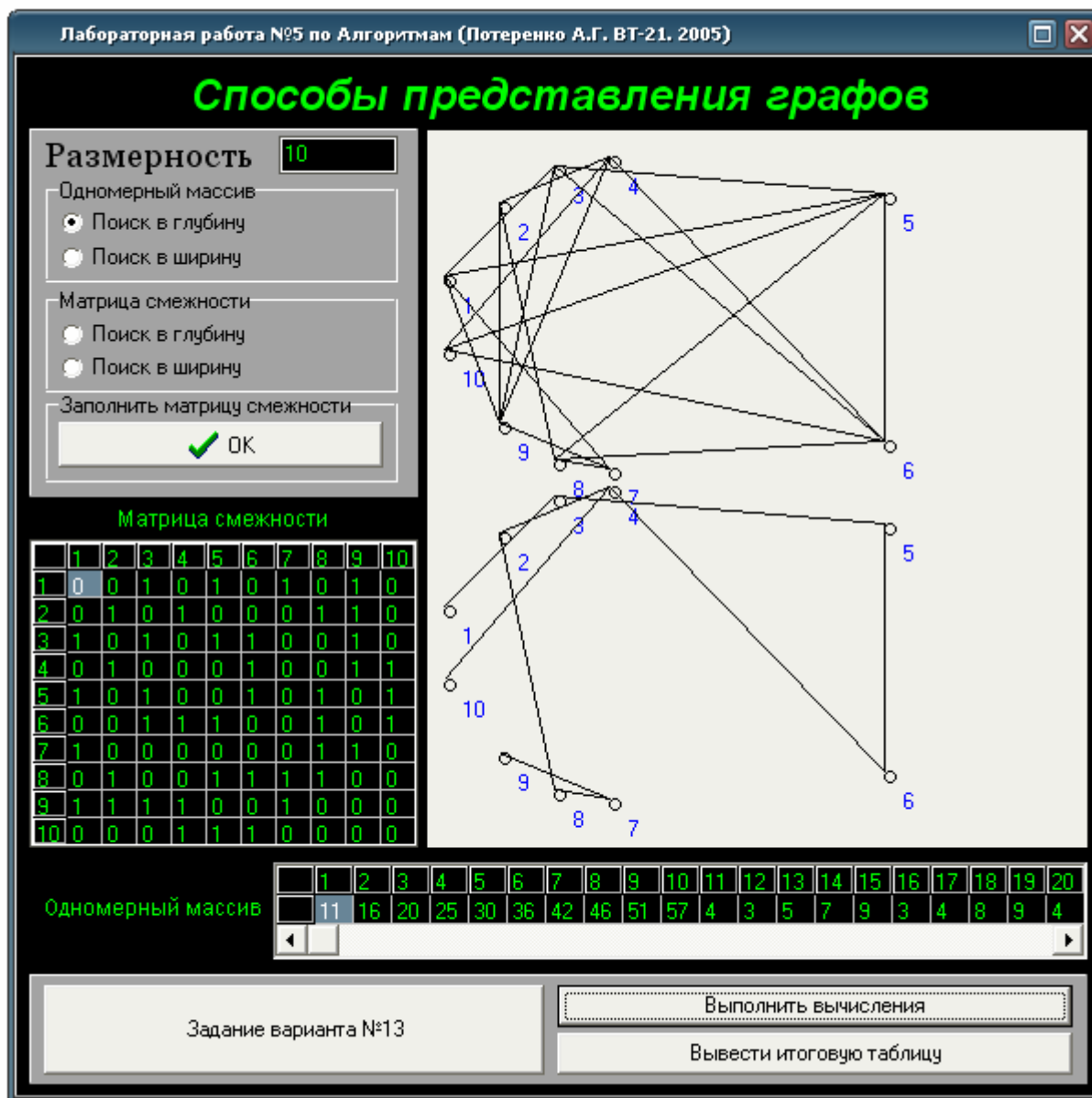
Пусть матрица имеет порядок n . Обход осуществляется нерекурсивной процедурой, то есть приходится создавать стек. Алгоритм состоит в том, чтобы создать очередь Q из n элементов и записывать в эту очередь в элемент $Q[i=\text{номер вершины}]$ единицу, если мы прошли элемент i . То есть, берем например, вершину 1. С ней смежные 2 и 4 вершины. Тогда очередь будет иметь вид $[1,1,0,1,0]$. Поиск продолжается до тех пор, пока очередь не заполнится полностью.

2.Обход в ширину.

Пусть матрица имеет порядок n . Обход осуществляется нерекурсивной процедурой. Берем первую вершину и в очереди Q помечаем ее двойкой, то есть в нее уже входить нельзя. Смотрим смежные с ней вершины и помечаем их 1-ми, как в обходе в глубину. Если вершина в очереди помечена как 1, то в нее входить можно. Так обходя все вершины, очередь заполняется 2 и 1. Обход продолжается до тех пор пока не будет 0 в очереди. Как только мы поместили 1 вершину, мы можем выбирать куда идти дальше – сколько 1-ц – столько и выбора. Идем в любую и помечаем ее 2 и т.д.

4. Описание и обоснование контрольных примеров.

Тестирование программы заключается в проверке дееспособности программы при количестве вершин от 3 до 10 при различных методах прохода: поиск в ширину и глубину по одномерному массиву, и то же самое с матрицей смежности. Контроль ввода также предусмотрен. На рисунке представлен фрагмент работы самой программы. Тестирование программы проводилось визуально, то есть при каждом значении порядка матрицы выводился граф и осто.



Итоговая таблица представлена на рисунке при одном из выборов обхода графа. В колонке представлено время обхода графа. Значение порядка может варьироваться в интервале от 3 до 5000. Тестирование также проводилось при различных видах обхода графа.

Итоговая таблица - время выполнения поиска остова при различных N=3..n

Количество вершин
 С 1000 По 1010
 Получить данные

Статус обработки данных

Вид обхода графа
☐ M/SM/GL ☒ M/SM/SH ☐ MAS/GL ☐ MAS/SH

Количество вершин	Поиск в глубину	Поиск в ширину
1000		0.15
1001		0.15
1002		0.16
1003		0.31
1004		1.16
1005		0.16
1006		0.15
1007		0.15
1008		0.31
1009		0.16
1010		0.16

5. Текст программы с комментариями (в виде приложения).

Исходный текст программы:

```

program Project1;

uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2},
  Unit3 in 'Unit3.pas' {Form3};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.Run;
end.

```

Исходный текст главного модуля программы:

```

unit Unit1;

interface

uses
  Windows, SysUtils, Variants, Classes, Graphics, Forms,
  Dialogs, StdCtrls, ExtCtrls, DB, DBTables, Grids, Buttons, Controls, Unit3;

type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;

```

```

Panel2: TPanel;
Button4: TButton;
Panel3: TPanel;
Button5: TButton;
Label2: TLabel;
Edit1: TEdit;
GroupBox1: TGroupBox;
GroupBox2: TGroupBox;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
RadioButton3: TRadioButton;
RadioButton4: TRadioButton;
Panel4: TPanel;
pbout: TPaintBox;
StG: TStringGrid;
GroupBox3: TGroupBox;
BitBtn1: TBitBtn;
Button6: TButton;
Label3: TLabel;
SG2: TStringGrid;
Label4: TLabel;
procedure FormCreate(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure RadioButton4Click(Sender: TObject);
procedure RadioButton1Click(Sender: TObject);
procedure RadioButton2Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Radio;
procedure RadioButton3Click(Sender: TObject);
procedure PROCQ(parametr:word);
procedure Graf_user(w:integer;parametr:word);
procedure GRAF_GLUBINA_MATR_SMEG;
procedure GRAF_SHIRINA_MATR_SMEG;
procedure CONVERT;
procedure Write_CONVERT;
procedure GRAF_GL_MAS;
procedure GRAF_Shir_MAS;
procedure GRAF_GLUBINA_MATR_SMEG_LES;

private
{ Private declarations }
public
{ Public declarations }
end;
const n=10;      //Максимальный размер матрицы
//-----10 вершин-----
maserx10:array[1..n] of word=(10,40,70,100,250,250,100,70,40,10);
masery10:array[1..n] of word=(80,40,20,15,35,170,185,180,160,120);
//-----9 вершин-----
maserx9:array[1..n] of word=(10,40,70,100,250,250,100,70,40,10);
masery9:array[1..n] of word=(80,40,20,15,35,170,185,180,160,120);
//-----8 вершин-----
maserx8:array[1..n] of word=(10,70,140,100,30,74,54,190,0,0);
masery8:array[1..n] of word=(60,40,80,130,130,85,133,36,0,0);
//-----7 вершин-----
maserx7:array[1..n] of word=(10,70,140,100,30,74,54,0,0,0);
masery7:array[1..n] of word=(60,40,80,130,130,85,115,0,0,0);
//-----6 вершин-----
maserx6:array[1..n] of word=(10,70,140,100,30,74,0,0,0,0);
masery6:array[1..n] of word=(80,40,80,130,130,85,0,0,0,0);
//-----5 вершин-----
maserx5:array[1..n] of word=(20,50,70,60,30,0,0,0,0,0);
masery5:array[1..n] of word=(40,10,40,70,70,0,0,0,0,0);
//-----4 вершин-----
maserx4:array[1..n] of word=(10,50,50,10,0,0,0,0,0,0);
masery4:array[1..n] of word=(10,10,60,60,0,0,0,0,0,0);
//-----3 вершин-----
maserx3:array[1..n] of word=(10,60,60,0,0,0,0,0,0,0);
masery3:array[1..n] of word=(50,20,80,0,0,0,0,0,0,0);

type os=record
    x:word;
    y:word;

```

```

        end;
var
    Form1: TForm1;
    M:array[1..n,1..n] of byte;           //Матрица смежности
    MC:array[1..n,1..n] of byte;         //Копия матрицы аналога-для восствновления
    M1:array[1..n,1..n] of byte;         //Матрица смежности
    Q:array[1..n] of integer;            //Массив занятости вершин
    Mas:array[1..10*n] of integer;       //Стек
    Msi:array[1..n*n+n] of integer;     //Одномерный массив
    flag,flag_glub,flad,flag_les:boolean;
    k,i,j,s,A,B,n1,j1:word;
    x1,y1,x2,y2,c,d,c1,d1,c2,d2,l:word;
    os_array:array[1..n] of os;
implementation

uses Unit2;

{$R *.dfm}

//-----
//                               Внутренняя процедура (+)
//-----
procedure TForm1.Graf_user(w:integer;parametr:word);
begin
    n1:=StrToInt(Edit1.Text);
    case n1 of
        3:begin
            for i:=1 to n1 do
                begin
                    os_array[i].x:=maserx3[i];
                    os_array[i].y:=masery3[i]+w;
                end;
            if parametr=1 then PROCQ(1) else PROCQ(2);
        end;
        4:begin
            for i:=1 to n1 do
                begin
                    os_array[i].x:=maserx4[i];
                    os_array[i].y:=masery4[i]+w;
                end;
            if parametr=1 then PROCQ(1) else PROCQ(2);
        end;
        5:begin
            for i:=1 to n1 do
                begin
                    os_array[i].x:=maserx5[i];
                    os_array[i].y:=masery5[i]+w;
                end;
            if parametr=1 then PROCQ(1) else PROCQ(2)
        end;
        6:begin
            for i:=1 to n1 do
                begin
                    os_array[i].x:=maserx6[i];
                    os_array[i].y:=masery6[i]+w;
                end;
            if parametr=1 then PROCQ(1) else PROCQ(2)
        end;
        7:begin
            for i:=1 to n1 do
                begin
                    os_array[i].x:=maserx7[i];
                    os_array[i].y:=masery7[i]+w;
                end;
            if parametr=1 then PROCQ(1) else PROCQ(2);
        end;
        8:begin
            for i:=1 to n1 do
                begin
                    os_array[i].x:=maserx8[i];
                    os_array[i].y:=masery8[i]+w;
                end;
            if parametr=1 then PROCQ(1) else PROCQ(2)
        end;
        9:begin

```



```

        for i:=1 to n1 do
            begin
                os_array[i].x:=maserx9[i];
                os_array[i].y:=masery9[i]+w;
            end;
        if parametr=1 then PROCQ(1) else PROCQ(2);
    end;
10:begin
    for i:=1 to n1 do
        begin
            os_array[i].x:=maserx10[i];
            os_array[i].y:=masery10[i]+w;
        end;
        if parametr=1 then PROCQ(1) else PROCQ(2)
    end;
end; //END CASE
end;
//-----
//                               Внутренняя процедура  (+)
//-----
procedure TForm1.PROCQ(parametr:word);
var par:word;
begin
    n1:=StrToInt(Edit1.Text);
    for i:=1 to n1 do
        begin
            x1:=os_array[i].x;
            y1:=os_array[i].y;
            c:=7;
            x2:=x1+c;
            y2:=y1+c;
            PbOut.Canvas.Ellipse(x1,y1,x2,y2);
            PbOut.Canvas.TextOut(x1+10,y1+10,VarToStr(i));
        end;
    for i:=1 to n1 do
        for j:=1 to n1 do
            begin
                case parametr of
                    1:par:=M[i,j];
                    2:par:=M1[i,j];
                end;
                if par=1 then
                    begin
                        c1:=os_array[i].x;
                        d1:=os_array[i].y;
                        c2:=os_array[j].x;
                        d2:=os_array[j].y;
                        pbout.Canvas.moveto(c1,d1);
                        pbout.Canvas.LineTo(c2,d2);
                    end;
            end;
        end;
    end;
end;
//-----
//                               Обход матрицы смежности в глубину (++)
//-----
procedure TForm1.GRAF_GLUBINA_MATR_SMEG;
label EXIT1;
begin
    n1:=StrToInt(Edit1.Text);
    for i:=1 to n do Q[i]:=0;
    for i:=1 to n do for j:=1 to n do M1[i,j]:=0;
    A:=1;
    j:=1;
    flag_glub:=false;
    REPEAT
        Q[A]:=1;
        for i:=1 to n1 do
            if (i<>A)and(M[A,i]=1)and(Q[i]<>1) then
                begin
                    M1[A,i]:=1;
                    Mas[j]:=A;    //Записываем вершину в стек
                    j:=j+1;
                    A:=i;
                    goto EXIT1;
                end;
        end;

```

```

        j:=j-1;    //Достаем из стека
        A:=Mas[j];
        EXIT1:
        if j=0 then flag_glub:=true;
    UNTIL flag_glub=true;
end;
//-----
//                               Обход матрицы смежности в ширину (++)
//-----
procedure TForm1.GRAF_SHIRINA_MATR_SMEG;
label ex;
begin
    n1:=StrToInt(Edit1.Text);
    for i:=1 to n do Q[i]:=0;
    A:=1;
    repeat
        Q[A]:=2;    {Признак того что эту вершину вообще нельзя использовать}
        for i:=1 to n1 do
            if i<>A then if M[A,i]=1 then
                begin
                    if Q[i]=1 then
                        begin
                            M[A,i]:=0;
                            M[i,A]:=0;
                        end;
                    if Q[i]<>2 then Q[i]:=1;
                end;

        s:=0;
        for i:=1 to n1 do if Q[i]=2 then s:=s+1;
        for i:=1 to n1 do if Q[i]=1 then
            begin
                A:=i;
                goto ex;
            end;

    ex:
    until s=n1; {Повторять до тех пор пока хотя бы одна из вершин Q[i] свободна}
end;
//-----
//                               Преобразуем граф из матрицы смежности в массив (++)
//-----
procedure TForm1.CONVERT;
begin
    n1:=StrToInt(Edit1.Text);
    SG2.ColCount:=n1*n1+n1+1;    //Вычисляем сколько колонок отображать
    k:=n1+1;
    l:=n1+2;
    for i:=1 to n1 do
        begin
            Msi[i]:=k;    //Запомнили указатель на элемент массива
            for j:=1 to n1 do
                if (j<>i)and(M[i,j]=1) then
                    begin
                        Msi[k]:=Msi[k]+1;
                        Msi[l]:=j;
                        l:=l+1;
                    end;

            k:=l;
            l:=l+1;
        end;

end;
//-----
//                               Дополнительная процедура (+)
//-----
procedure TForm1.Write_CONVERT;
begin
    n1:=StrToInt(Edit1.Text);
    for i:=1 to (n1*n1+n1) do
        begin
            SG2.Cells[i,0]:=VarToStr(i);
            SG2.Cells[i,1]:=VarToStr(Msi[i]);
        end;

end;
//-----
//                               Обход одномерного массива в глубину (++)
//-----

```

```

procedure TForm1.GRAF_GL_MAS;
label ex;
var j1:word;
begin
  n1:=StrToInt(Edit1.Text);
  for i:=1 to n1 do Q[i]:=0;
  for i:=1 to n do for j:=1 to n do M1[i,j]:=0;
  i:=1;
  j1:=1;
  Q[i]:=1;
  REPEAT
    A:=Msi[i];          //A=указатель на элемент последовательности вершины
    B:=Msi[A];          //B=число смежных вершин у данной вершины
    if B=0 then
      begin
        Q[i]:=1;
        M1[i,Mas[j1]]:=1;
      end
    else
      for j:=1 to B do //Проходим смежные вершины и помечаем как занятые
        if Q[Msi[A+j]]=0 then
          begin
            Mas[j1]:=i;    //Записали в стек
            j1:=j1+1;
            M1[i,Msi[A+j]]:=1; //Пометили как занятую в матрице
            Q[Msi[A+j]]:=1;   //Пометили в очереди
            i:=Msi[A+j];
            goto ex;
          end;
        j1:=j1-1;
        i:=Mas[j1];
      ex:
      UNTIL j1=0;
end;
//-----
//                               Обход одномерного массива в ширину
//-----
procedure TForm1.GRAF_Shir_MAS;
label ex;
begin
  n1:=StrToInt(Edit1.Text);
  for i:=1 to n do Q[i]:=0;
  for i:=1 to n do for j:=1 to n do M1[i,j]:=0;
  i:=1;
  Q[i]:=2;    //В нее нельзя входить и она занята
  //-----
  k:=n1;
  REPEAT
    A:=Msi[i];          //A=указатель на элемент последовательности вершины
    B:=Msi[A];          //B=число смежных вершин у данной вершины
    if B<>0 then
      begin
        for j:=1 to B do //Проходим смежные вершины и помечаем как занятые
          if Q[Msi[A+j]]=0 then
            begin
              M1[i,Msi[A+j]]:=1;
              Q[Msi[A+j]]:=1;
            end;
        for i:=1 to n1 do if Q[i]=1 then
          begin
            Q[i]:=2;
            A:=i;
            k:=k-1;
            goto ex;
          end;
        ex:
      end;
    UNTIL k=1;
end;
//-----
//                               Обработчик кнопки "Выполнить вычисления" (+)
//-----
procedure TForm1.Button5Click(Sender: TObject);
begin
  try

```

```

if (Edit1.Text<>'')and(StrToInt(Edit1.Text)>2)and(StrToInt(Edit1.Text)<=10) then
  if flad=false then //Пользователь может переполнить буфер
    begin
      //-----Очищаем канву-----
      PbOut.Color:=ClBtnFace;
      PbOut.Canvas.Rectangle(0,0,377,393);
      Graf_user(0,1); //Выводим граф
      //-----
      if RadioButton3.Checked=true then
        begin
          GRAF_GLUBINA_MATR_SMEG;
          Graf_user(180,2); //Выводим остов
        end;
      if RadioButton4.Checked=true then
        begin
          GRAF_SHIRINA_MATR_SMEG;
          Graf_user(180,1); //Выводим остов
        end;
      if RadioButton1.Checked=true then
        begin
          CONVERT;//Преобразуем матрицу смежности в одномерный массив
          Write_CONVERT; //Показываем одномерный массив
          GRAF_GL_MAS;
          Graf_user(180,2); //Выводим остов
        end;
      if RadioButton2.Checked=true then
        begin
          CONVERT;//Преобразуем матрицу смежности в одномерный массив
          Write_CONVERT; //Показываем одномерный массив
          GRAF_Shir_MAS;
          Graf_user(180,2); //Выводим остов
        end;
    end;
except
end;
flad:=true; //Обработка ошибки повторного нажатия
end;
//-----
// Заполнение матрицы смежности (+)
//-----
procedure TForm1.GRAF_GLUBINA_MATR_SMEG_LES;
label EXIT1;
begin
  n1:=StrToInt(Edit1.Text);
  for i:=1 to n do Q[i]:=0;
  A:=1;
  j:=1;
  flag_glub:=false;
  REPEAT
    Q[A]:=1;
    for i:=1 to n1 do
      if (i<>A)and(M[A,i]=1)and(Q[i]<>1) then
        begin
          Mas[j]:=A; //Записываем вершину в стек
          j:=j+1;
          A:=i;
          goto EXIT1;
        end;
    j:=j-1; //Достаем из стека
    A:=Mas[j];
  EXIT1:
    if j=0 then flag_glub:=true;
  UNTIL flag_glub=true;
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  try
    n1:=StrToInt(Edit1.Text);
    if (n1>2)and(n1<=10) then
      begin
        flad:=false;
        for i:=1 to n do
          begin
            for j:=1 to n do
              begin

```

```

M[i,j]:=0;
M1[i,j]:=0;
STG.Cells[i,j]:='';
end;

Mas[i]:=0;
Q[i]:=0;
end;

for i:=1 to n*n+n do Msi[i]:=0;
SG2.ColCount:=2;
SG2.Cells[1,1]:='';
SG2.Cells[1,0]:='';
//-----
Randomize;
REPEAT
  for i:=1 to n1 do for j:=i to n1 do
    begin
      M[i,j]:=random(2);
      M[j,i]:=M[i,j];
      MC[i,j]:=M[i,j];
      MC[j,i]:=M[j,i];
    end;
    flag_les:=false;
    GRAF_GLUBINA_MATR_SMEG_LES;
    for i:=1 to n1 do if Q[i]=0 then flag_les:=true;
      if flag_les=true then for i:=1 to n*n+n do Msi[i]:=0;
    UNTIL flag_les=false;
    for i:=1 to n1 do for j:=1 to n1 do
      STG.Cells[j,i]:=VarToStr(M[i,j]);
    end;
except
  //Не выводим никаких сообщений
end;
end;
//-----
//                               Вывод итоговой таблицы      (+)
//-----
procedure TForm1.Button6Click(Sender: TObject);
begin
  Form2.PB1.Position:=0;
  Form2.SG1.RowCount:=2;
  Form2.SG2.RowCount:=2;
  Form2.SG1.Cells[0,1]:='';
  Form2.SG2.Cells[1,1]:='';
  Form2.SG1.Cells[2,1]:='';
  Form2.SG2.Cells[0,1]:='';
  Form2.SG1.Cells[1,1]:='';
  Form2.SG2.Cells[2,1]:='';
  Form2.Edit2.Text:='';
  Form2.Edit1.Text:='';
  Form2.Left:=193;
  Form2.Top:=196;
  Form2.Show;           //Показали итоговую таблицу
end;
//-----
//                               Инициализация      (+)
//-----
procedure TForm1.FormCreate(Sender: TObject);
begin
  SG2.ColCount:=2;           //Одномерный массив
  RadioButton1.Checked:=true;
  Form1.Left:=229;
  Form1.Top:=113;
  PbOut.Font.Color:=ClBlue;
  for i:=1 to n do
    begin
      STG.Cells[0,i]:=VarToStr(i);
      STG.Cells[i,0]:=VarToStr(i);
    end;
end;
//-----
//                               Переключение компонентов RADIOBUTTON (+)
//-----
procedure TForm1.Radio;
begin
  n1:=StrToInt(Edit1.Text);

```

```

    for i:=1 to n do
    begin
        for j:=1 to n do
            begin
                M[i,j]:=MC[i,j];
                M1[i,j]:=0;
            end;
        Q[i]:=0;
    end;
    flad:=false;
    for i:=1 to n*n+n do Msi[i]:=0;
    SG2.ColCount:=2;SG2.Cells[1,1]:='';SG2.Cells[1,0]:='';
end;
procedure TForm1.RadioButton4Click(Sender: TObject);
begin
    try
        Radio;
    except
    end;
    RadioButton1.Checked:=false;
    RadioButton2.Checked:=false;
    RadioButton3.Checked:=false;
end;
procedure TForm1.RadioButton1Click(Sender: TObject);
begin
    try
        Radio;
    except
    end;
    RadioButton2.Checked:=false;
    RadioButton3.Checked:=false;
    RadioButton4.Checked:=false;
end;
procedure TForm1.RadioButton2Click(Sender: TObject);
begin
    try
        Radio;
    except
    end;
    RadioButton1.Checked:=false;
    RadioButton3.Checked:=false;
    RadioButton4.Checked:=false;
end;
procedure TForm1.RadioButton3Click(Sender: TObject);
begin
    try
        Radio;
    except
    end;
    RadioButton1.Checked:=false;
    RadioButton2.Checked:=false;
    RadioButton4.Checked:=false;
end;
//-----
//                               Задание варианта №13 (+)
//-----
procedure TForm1.Button4Click(Sender: TObject);
begin
    Form3.Left:=278;
    Form3.Top:=253;
    Form3.ShowModal;
end;
end.

```

Исходный текст дополнительного модуля программы:

```

unit Unit2;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Grids, StdCtrls, ExtCtrls, ComCtrls;

```

```

type
  TForm2 = class(TForm)
    SG1: TStringGrid;
    SG2: TStringGrid;
    Panel1: TPanel;
    Panel2: TPanel;
    GroupBox1: TGroupBox;
    Button1: TButton;
    Edit1: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    Edit2: TEdit;
    PB1: TProgressBar;
    Label5: TLabel;
    GroupBox2: TGroupBox;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure GRAF_GLUBINA_MATR_SMEG;
    procedure GRAF_SHIRINA_MATR_SMEG;
    procedure CONVERT;
    procedure GRAF_GL_MAS;
    procedure GRAF_Shir_MAS;

  private
    { Private declarations }
  public

    { Public declarations }
  end;
const n=5000;
var
  Form2: TForm2;
  M:array[1..n,1..n] of byte;    //Матрица смежности
  MC:array[1..n,1..n] of byte;  //Копия матрицы аналога-для восствновления
  M1:array[1..n,1..n] of byte;  //Матрица смежности
  Q:array[1..n] of byte;        //Массив занятости вершин
  Mas:array[1..10000] of longint; //Стек
  Msi:array[1..n*n+n] of longint;//Одномерный массив
  flag_les, flag_glub:boolean;
  A,i,p,fil,fi2,fi3,fi4,j,li,g,mil,sec,delta,deltas,j1,n1,s,B,k,l:longint;
  GLT:TSystemTime;
  ki:longint;

implementation

{$R *.dfm}
//-----Инициализация-----
procedure TForm2.FormCreate(Sender: TObject);
begin
  SG1.Cells[0,0]:='Количество вершин';
  SG2.Cells[0,0]:='Количество вершин';
  SG1.Cells[1,0]:='Поиск в глубину';
  SG2.Cells[1,0]:='Поиск в глубину';
  SG1.Cells[2,0]:='Поиск в ширину';
  SG2.Cells[2,0]:='Поиск в ширину';
end;
//-----
//
//-----
procedure TForm2.GRAF_GLUBINA_MATR_SMEG;
label EXIT1;
var j,i:longint;
begin
  A:=1;
  j:=1;
  flag_glub:=false;
  REPEAT
    Q[A]:=1;
    for i:=1 to n1 do
      if (i<>A)and(M[A,i]=1)and(Q[i]<>1) then

```

```

begin
    M1[A,i]:=1;
    Mas[j]:=A;    //Записываем вершину в стек
    j:=j+1;
    A:=i;
    goto EXIT1;
end;
j:=j-1;    //Достаем из стека
A:=Mas[j];
EXIT1:
    if j=0 then flag_glub:=true;
UNTIL flag_glub=true;
end;
procedure TForm2.GRAF_SHIRINA_MATR_SMEG;
label ex;
var i:longint;
begin
    A:=1;
    repeat
        Q[A]:=2;    {Признак того что эту вершину вообще нельзя использовать}
        for i:=1 to n1 do
            if i<>A then if M[A,i]=1 then
                begin
                    if Q[i]=1 then
                        begin
                            M[A,i]:=0;
                            M[i,A]:=0;
                        end;
                    if Q[i]<>2 then Q[i]:=1;
                end;
            s:=0;
            for i:=1 to n1 do if Q[i]=2 then s:=s+1;
            for i:=1 to n1 do if Q[i]=1 then
                begin
                    A:=i;
                    goto ex;
                end;
            ex:
            until s=n1;    {Повторять до тех пор пока хотя бы одна из вершин Q[i] свободна}
        end;
    procedure TForm2.CONVERT;
    var i,j:longint;
    begin
        k:=n1+1;
        l:=n1+2;
        for i:=1 to n1 do
            begin
                Msi[i]:=k;    //Запомнили указатель на элемент массива
                for j:=1 to n1 do
                    if (j<>i)and(M[i,j]=1) then
                        begin
                            Msi[k]:=Msi[k]+1;
                            Msi[l]:=j;
                            l:=l+1;
                        end;
                k:=l;
                l:=l+1;
            end;
        end;
    procedure TForm2.GRAF_GL_MAS;
    label ex;
    var j1,j,i:longint;
    begin
        i:=1;
        j1:=1;
        Q[i]:=1;
        REPEAT
            A:=Msi[i];    //A=указатель на элемент последовательности вершины
            B:=Msi[A];    //B=число смежных вершин у данной вершины
            if B=0 then
                begin
                    Q[i]:=1;
                    M1[i,Mas[j1]]:=1;
                end
            else

```



```

        for j:=1 to B do //Проходим смежные вершины и помечаем как занятые
            if Q[Msi[A+j]]=0 then
                begin
                    Mas[j1]:=i; //Записали в стек
                    j1:=j1+1;
                    M1[i,Msi[A+j]]:=1; //Пометили как занятую в матрице
                    Q[Msi[A+j]]:=1; //Пометили в очереди
                    i:=Msi[A+j];
                    goto ex;
                end;

            j1:=j1-1;
            i:=Mas[j1];
            ex:
        UNTIL j1=0;
    end;
procedure TForm2.GRAF_Shir_MAS;
label ex;
var i,j:longint;
begin
    i:=1;
    Q[i]:=2; //В нее нельзя входить и она занята
    //-----
    k:=n1;
    REPEAT
        A:=Msi[i]; //A=указатель на элемент последовательности вершины
        B:=Msi[A]; //B=число смежных вершин у данной вершины
        if B<>0 then
            begin
                for j:=1 to B do //Проходим смежные вершины и помечаем как занятые
                    if Q[Msi[A+j]]=0 then
                        begin
                            M1[i,Msi[A+j]]:=1;
                            Q[Msi[A+j]]:=1;
                        end;
                for i:=1 to n1 do if Q[i]=1 then
                    begin
                        Q[i]:=2;
                        A:=i;
                        k:=k-1;
                        goto ex;
                    end;
                ex:
            end;
        UNTIL k=1;
    end;
    //-----
    // Выполнение всех операций
    //-----
procedure TForm2.Button1Click(Sender: TObject);
procedure Copy_MC_M;
var i,j:longint;
begin
    for i:=1 to n1 do
        for j:=1 to n1 do
            M[i,j]:=MC[i,j];
end;
procedure GRAF_GLUBINA_MATR_SMEG_LES;
label EXIT1;
var i,j:longint;
begin
    for i:=1 to n1 do Q[i]:=0;
    A:=1;
    j:=1;
    flag_glub:=false;
    REPEAT
        Q[A]:=1;
        for i:=1 to n1 do
            if (i<>A)and(M[A,i]=1)and(Q[i]<>1) then
                begin
                    Mas[j]:=A; //Записываем вершину в стек
                    j:=j+1;
                    A:=i;
                    goto EXIT1;
                end;
        j:=j-1; //Достаем из стека

```

```

        A:=Mas[j];
        EXIT1:
        if j=0 then flag_glub:=true;
    UNTIL flag_glub=true;
end;
//-----
//                               Основной блок программы
//-----
begin
    SG1.RowCount:=1; //Количество строк
    SG2.RowCount:=1; //Количество строк
    try
        if (StrToInt(Edit2.Text)<=n)and(StrToInt(Edit1.Text)>=3)and
            (StrToInt(Edit2.Text)>=StrToInt(Edit1.Text))and((RadioButton1.Checked=true)or
            (RadioButton2.Checked=true)or(RadioButton3.Checked=true)or
            (RadioButton4.Checked=true)) then
            begin
                //-----
                //-----
                //-----
                PB1.Position:=0;
                g:=StrToInt(Edit2.Text)-StrToInt(Edit1.Text);
                for i:=1 to g+1 do
                    begin
                        SG1.RowCount:=SG1.RowCount+1;
                        SG2.RowCount:=SG2.RowCount+1;
                        SG1.Cells[0,i]:=VarToStr(i+StrToInt(Edit1.Text)-1);
                        SG2.Cells[0,i]:=VarToStr(i+StrToInt(Edit1.Text)-1);
                    end;
                //-----
                p:=1;
                fi1:=1;
                fi2:=1;
                fi3:=1;
                fi4:=1;
                for ki:=StrToInt(Edit1.Text) to StrToInt(Edit2.Text) do
                    begin
                        //-----
                        n1:=ki; //Число вершин в графе
                        Randomize;
                        for i:=1 to n do Q[i]:=0;
                        REPEAT
                            for i:=1 to n1 do for j:=i to n1 do
                                begin
                                    M[i,j]:=random(2);
                                    M[j,i]:=M[i,j];
                                    //Копируем M in MC
                                    MC[i,j]:=M[i,j];
                                    MC[j,i]:=M[j,i];
                                end;
                            flag_les:=false;
                            GRAF_GLUBINA_MATR_SMEG_LES;
                            for i:=1 to n1 do if Q[i]=0 then flag_les:=true;
                            if flag_les=true then for i:=1 to n1*n1+1 do Msi[i]:=0;
                        UNTIL flag_les=false;
                        //-----
                        if RadioButton1.Checked=true then
                            begin
                                for i:=1 to n1 do Q[i]:=0;
                                GetLocalTime(GLT);
                                mil:=GLT.wMilliseconds;
                                sec:=GLT.wSecond;
                                //-----
                                GRAF_GLUBINA_MATR_SMEG;
                            end;
                        if RadioButton2.Checked=true then
                            begin
                                for i:=1 to n1 do Q[i]:=0;
                                GetLocalTime(GLT);
                                mil:=GLT.wMilliseconds;
                                sec:=GLT.wSecond;
                                //-----
                                GRAF_SHIRINA_MATR_SMEG;
                            end;
                        if RadioButton3.Checked=true then

```

```

begin
    for i:=1 to n1 do Q[i]:=0;
    CONVERT;
    GetLocalTime(GLT);
    mil:=GLT.wMilliseconds;
    sec:=GLT.wSecond;
    //-----
    GRAF_GL_MAS;
end;
if RadioButton4.Checked=true then
begin
    for i:=1 to n1 do Q[i]:=0;
    CONVERT;
    GetLocalTime(GLT);
    mil:=GLT.wMilliseconds;
    sec:=GLT.wSecond;
    //-----
    GRAF_Shir_MAS;
end;
//-----Измерили время-----
GetLocalTime(GLT);
if (GLT.wSecond-sec)>=0 then deltas:=GLT.wSecond-sec
else
begin
    deltas:=60-sec;
    deltas:=deltas+GLT.wSecond;
end;
if (GLT.wMilliseconds-mil)>=0 then delta:=GLT.wMilliseconds-mil
else
begin
    delta:=1000-mil;
    delta:=delta+GLT.wMilliseconds;
end;
//-----
if RadioButton1.Checked=true then
begin
    SG1.Cells[1,fi1]:=VarToStr(deltas)+'.'+VarToStr(delta);
    fi1:=fi1+1;
end;
if RadioButton2.Checked=true then
begin
    SG1.Cells[2,fi2]:=VarToStr(deltas)+'.'+VarToStr(delta);
    fi2:=fi2+1;
end;
if RadioButton3.Checked=true then
begin
    SG2.Cells[1,fi3]:=VarToStr(deltas)+'.'+VarToStr(delta);
    fi3:=fi3+1;
end;
if RadioButton4.Checked=true then
begin
    SG2.Cells[2,fi4]:=VarToStr(deltas)+'.'+VarToStr(delta);
    fi4:=fi4+1;
end;
//-----Восстановили матрицу M-----
Copy_MC_M;
PB1.Position:=10*p;
if ki=p*round((10*(StrToInt(Edit2.Text)-StrToInt(Edit1.Text)))/100)+StrToInt(Edit1.Text)
then p:=p+1;
//-----
end;
PB1.Position:=100;
//-----
//-----
//-----
end;
except //Пустой оператор обработки исключения
end;
end;
end.

```