

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
Федерального государственного бюджетного образовательного учреждения
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

ЛАБОРАТОРНАЯ РАБОТА №3

По Дискретной математике

Руководитель

Кульков Я.Ю.

(фамилия, инициалы)

(подпись)

(дата)

Студент _____ ПИН - 121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись)

(дата)

Муром 2022

Лабораторная работа №3

Тема: ИЗУЧЕНИЕ АЛГОРИТМА ОБХОДА ГРАФА

Цель работы: Изучить способы описания и представления в ЭВМ графов, а также базовый алгоритм поиска путей в графе.

Порядок выполнения работы

1. Составить программу, осуществляющую чтение матрицы смежности
2. Реализовать алгоритм обхода графа в ширину для поиска путей
3. Программа должна выводить на экран номера смежных вершин в порядке обхода от начальной, вводимой пользователем. Также выводить длину каждого пути.

					МИ ВлГУ 09.03.04			
Изм.	Лист	№ докум.	Подпись	Дата	ИЗУЧЕНИЕ АЛГОРИТМА ОБХОДА ГРАФА	Лит.	Лист	Листов
Разраб.		Ермилов М.В.						
Провер.		Кульков Я.Ю.					2	5
Реценз.						ПИН-121		
Н. Контр.								
Утверд.								

Код:

Класс Graph:

```
class Graph
{
    private int[,] adjacency;
    public int numEdges;
    public Graph(string file)
    {
        StreamReader r = new StreamReader(file);
        string strjson = r.ReadToEnd();
        JObject json = JObject.Parse(strjson);
        numEdges = (int)json["Edges"];
        adjacency = new int[numEdges, numEdges];
        for(int i = 0; i < numEdges; i++)
            for (int j = 0; j < numEdges; j++)
                adjacency[i, j] = (int)json["Adjacency"][i][j];
    }
    public Graph() => numEdges = 0;

    public string BFS(int from, int to)
    {
        if (numEdges <= 0)
            return "нет пути";
        int[] mark = new int[numEdges];
        int[] parent = new int[numEdges];
        for (int i = 0; i < numEdges; i++)
        {
            mark[i] = 0;
            parent[i] = from;
        }
        Queue<int> Q = new Queue<int>();
        int v = from;
        mark[v] = 1;
        Q.Enqueue(v);
        while (Q.Count != 0)
        {
            v = Q.Dequeue();
            for (int i = 0; i < numEdges; i++)
                if ((adjacency[v, i] != 0) && (mark[i] == 0))
                {
                    mark[i] = 1;
                    Q.Enqueue(i);
                    parent[i] = v;
                }
            mark[v] = 2;
        }

        string ret = to.ToString();
        int check = to;
        while (true)
        {
            check = parent[check];
            ret = check + "-" + ret;
            if (check == from)
                break;
        }
        return ret;
    }
}
```

Класс Form1:

```
public partial class Form1 : Form
{
    private Graph g;
    private int from { get => edges(TextBoxFrom); }
    private int to { get => edges(TextBoxTo); }
    private void ButtonTracer_Click(object sender, EventArgs e) => LabelTracer.Text
= $"Путь из {from} в {to}: {g.BFS(from, to)}";
    public Form1()
    {
        g = new Graph();
        InitializeComponent();
    }
    private int edges(TextBox textBox)
    {
        int r = 0;
        int.TryParse(textBox.Text, out r);
        if (r < 0)
            r = 0;
        if (r >= g.numEdges)
            r = g.numEdges - 1;
        return r;
    }
    private void ButtonFile_Click(object sender, EventArgs e)
    {
        OpenFileDialog OPF = new OpenFileDialog();
        OPF.Filter = "Файлы json|*.json";
        if (OPF.ShowDialog() == DialogResult.OK)
        {
            g = new Graph(OPF.FileName);
        }
    }
}
```

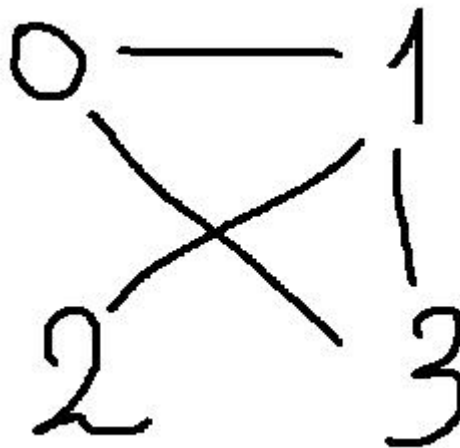


Рис 1 - Граф в тестовом json файле

Содержание тестового json файла:

```
{
  "Edges": 4,
  "Adjacency": [
    [0, 1, 0, 1],
    [1, 0, 1, 1],
    [0, 1, 0, 0],
    [1, 1, 0, 0]
  ],
}
```

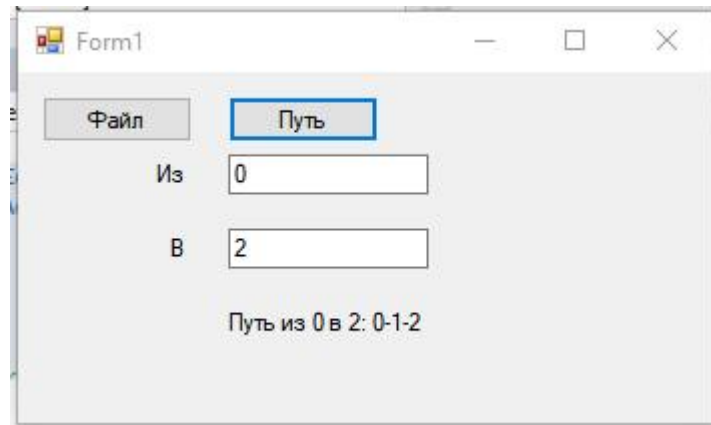


Рис 2 - пример работы программы