ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт – Петербургский государственный университет телекоммуникаций

им. проф. М.А. Бонч-Бруевича»

КАФЕДРА ИНФОРМАЦИОННЫХ УПРАВЛЯЮЩИХ СИСТЕМ

**Лабораторно-практическая работа № 01**

по дисциплине

**«СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ»**

Студент гр. 831 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пономарев Е.И.

(подпись)

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Параничев А.В.

(оценка и подпись)

Санкт-Петербург

2021 год

**Лабораторно-практическая работа № 01: Представление сети с помощью связанного списка**

Выполним представление сети из 5 узлов и 4 ветвей, используя онлайн-компилятор <https://www.onlinegdb.com/> для следующего кода на языке C# для варианта 83114:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text.RegularExpressions;

namespace Main

{

using s\_network = Struct\_SimpeNetwork.SimpleNetwork;

using s\_vertex = Struct\_SimpeNetwork.SimpleNetwork.SimpleVertex;

using s\_edge = Struct\_SimpeNetwork.SimpleNetwork.SimpleEdge;

public struct Struct\_SimpeNetwork

{

public struct SimpleNetwork

{

// declare arrays and data

public IList<SimpleVertex> arr\_vertexes;

public IList<SimpleEdge> arr\_edges;

public int num\_of\_vertexes;

public int num\_of\_edges;

// structure for vertex //VVVVVVVVVVVVVVVVVVVVVVVVVVVV

public struct SimpleVertex //VVVVVVVVVVVVVVVVVVVVVVVVV

{

// struct data

public int number;

public ConsoleColor color;

public int x\_coord, y\_coord;

} //VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV

// structure for edge //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

public struct SimpleEdge //EEEEEEEEEEEEEEEEEEEEEEEEEEE

{

// struct data

public SimpleVertex begin\_SimpleVertex;

public SimpleVertex end\_SimpleVertex;

public ConsoleColor color;

// struct constructor

} //EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

}

}

public class Program

{

public static void Main(string[] args)

{

int Num = 83114;

// initializing the network

s\_network netw = new s\_network();

netw.arr\_vertexes = new List<s\_vertex>();

netw.arr\_edges = new List<s\_edge>();

// Task 1)

netw.num\_of\_vertexes = 5;

netw.num\_of\_edges = netw.num\_of\_vertexes-1;

// define array's data

for(int num\_i = 0; num\_i < netw.num\_of\_vertexes; num\_i++)

{ // vertexes are set at the diagonal line

// (узлы расставляются по диагонали

s\_vertex new\_v = new s\_vertex(); //(num\_i, ConsoleColor.Black, num\_i\*10, num\_i\*10);

new\_v.number = num\_i;

new\_v.color = (ConsoleColor)num\_i;

new\_v.x\_coord = (Num+2\*num\_i)%21;

new\_v.y\_coord = (Num+2\*num\_i)%22;

netw.arr\_vertexes.Add(new\_v);

}

for(int num\_i = 0; num\_i < netw.num\_of\_edges; num\_i++)

{

// edges are concatenated the vertexes by diagonal

// (ветви соединяют соответствующие узлы по диагонали: (0->1), (1->2), (2->3), (3->4))

s\_edge new\_ed = new s\_edge();

new\_ed.begin\_SimpleVertex = netw.arr\_vertexes[num\_i];

new\_ed.end\_SimpleVertex = netw.arr\_vertexes[num\_i + 1];

new\_ed.color = new\_ed.begin\_SimpleVertex.color;

netw.arr\_edges.Add(new\_ed);

}

OutNetworkData(netw);

// Task 2)

for(int num\_i = 0; num\_i < netw.num\_of\_vertexes; num\_i++)

{

s\_vertex v = netw.arr\_vertexes[num\_i];

v.x\_coord = ((Num + 100\*num\_i)%31)/2;

v.y\_coord = (Num + 100\*num\_i)%22;

netw.arr\_vertexes[num\_i] = v;

}

OutNetworkData(netw);

// Task 3

for(int num\_i = 0; num\_i < netw.num\_of\_edges; num\_i++)

{

s\_edge ed = netw.arr\_edges[num\_i];

ed.begin\_SimpleVertex = netw.arr\_vertexes[(Num / (num\_i+1))%netw.num\_of\_vertexes];

ed.end\_SimpleVertex = netw.arr\_vertexes[(Num + num\_i\*num\_i)%netw.num\_of\_vertexes];

ed.color = ed.begin\_SimpleVertex.color;

netw.arr\_edges[num\_i] = ed;

}

OutNetworkData(netw);

}

public static void OutNetworkData(s\_network network)

{

// output data

string str = "Vertex'es data:";

Console.WriteLine(str);

for(int num\_i = 0; num\_i < network.num\_of\_vertexes; num\_i++)

{ // output all vertex data

str = "number = " + network.arr\_vertexes[num\_i].number.ToString() +

", Color = " + network.arr\_vertexes[num\_i].color.ToString() +

", x = " + network.arr\_vertexes[num\_i].x\_coord.ToString() +

", y = " + network.arr\_vertexes[num\_i].y\_coord.ToString();

Console.WriteLine(str);

}

str = "Edge's data:";

Console.WriteLine(str);

for(int num\_i = 0; num\_i < network.num\_of\_edges; num\_i++)

{ // output all edges data

str = "index " + num\_i.ToString() + ": begin\_Vertex = {number = " +

network.arr\_edges[num\_i].begin\_SimpleVertex.number.ToString() +

", color = " + network.arr\_edges[num\_i].begin\_SimpleVertex.color.ToString() +

"},\n end\_Vertex = {number = " + network.arr\_edges[num\_i].end\_SimpleVertex.number.ToString() +

", color = " + network.arr\_edges[num\_i].end\_SimpleVertex.color.ToString() +

"}, edge's color = " + network.arr\_edges[num\_i].color.ToString();

Console.WriteLine(str);

}

//Console.ReadKey();

}

}

}

В результате работы программы получим (https://onlinegdb.com/ZqVEuPWfU):

Vertex'es data:

number = 0, Color = Black, x = 17, y = 20

number = 1, Color = DarkBlue, x = 19, y = 0

number = 2, Color = DarkGreen, x = 0, y = 2

number = 3, Color = DarkCyan, x = 2, y = 4

number = 4, Color = DarkRed, x = 4, y = 6

Edge's data:

index 0: begin\_Vertex = {number = 0, color = Black},

end\_Vertex = {number = 1, color = DarkBlue}, edge's color = Black

index 1: begin\_Vertex = {number = 1, color = DarkBlue},

end\_Vertex = {number = 2, color = DarkGreen}, edge's color = DarkBlue

index 2: begin\_Vertex = {number = 2, color = DarkGreen},

end\_Vertex = {number = 3, color = DarkCyan}, edge's color = DarkGreen

index 3: begin\_Vertex = {number = 3, color = DarkCyan},

end\_Vertex = {number = 4, color = DarkRed}, edge's color = DarkCyan

Vertex'es data:

number = 0, Color = Black, x = 1, y = 20

number = 1, Color = DarkBlue, x = 5, y = 10

number = 2, Color = DarkGreen, x = 8, y = 0

number = 3, Color = DarkCyan, x = 12, y = 12

number = 4, Color = DarkRed, x = 0, y = 2

Edge's data:

index 0: begin\_Vertex = {number = 0, color = Black},

end\_Vertex = {number = 1, color = DarkBlue}, edge's color = Black

index 1: begin\_Vertex = {number = 1, color = DarkBlue},

end\_Vertex = {number = 2, color = DarkGreen}, edge's color = DarkBlue

index 2: begin\_Vertex = {number = 2, color = DarkGreen},

end\_Vertex = {number = 3, color = DarkCyan}, edge's color = DarkGreen

index 3: begin\_Vertex = {number = 3, color = DarkCyan},

end\_Vertex = {number = 4, color = DarkRed}, edge's color = DarkCyan

Vertex'es data:

number = 0, Color = Black, x = 1, y = 20

number = 1, Color = DarkBlue, x = 5, y = 10

number = 2, Color = DarkGreen, x = 8, y = 0

number = 3, Color = DarkCyan, x = 12, y = 12

number = 4, Color = DarkRed, x = 0, y = 2

Edge's data:

index 0: begin\_Vertex = {number = 4, color = DarkRed},

end\_Vertex = {number = 4, color = DarkRed}, edge's color = DarkRed

index 1: begin\_Vertex = {number = 2, color = DarkGreen},

end\_Vertex = {number = 0, color = Black}, edge's color = DarkGreen

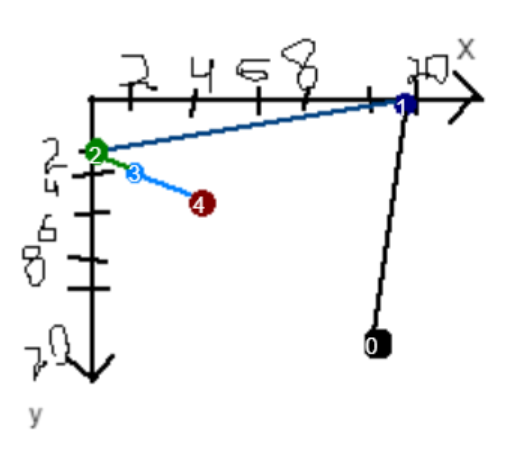
index 2: begin\_Vertex = {number = 4, color = DarkRed},

end\_Vertex = {number = 3, color = DarkCyan}, edge's color = DarkRed

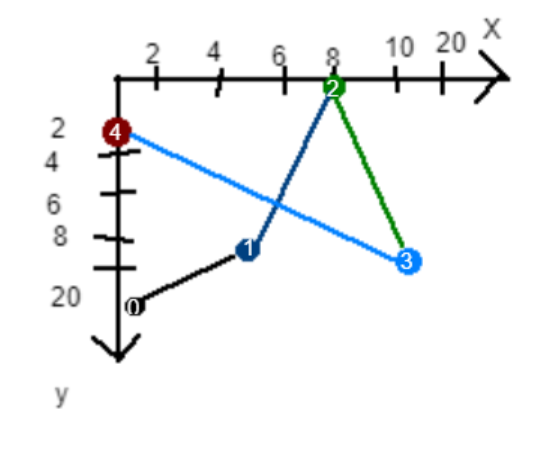
index 3: begin\_Vertex = {number = 3, color = DarkCyan},

end\_Vertex = {number = 3, color = DarkCyan}, edge's color = DarkCyan

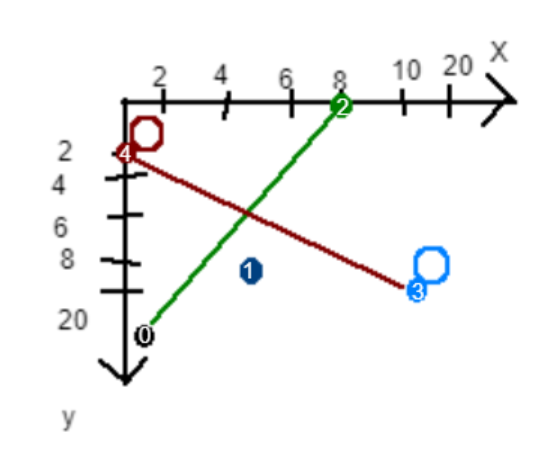
Представим графически 3 графа сети (Task\_1, Task\_2 и Task\_3 в коде), представленных в данном коде (рис.1-3).



*Рис.1. Представление графа сети в результате выполнения задания Task\_1*



*Рис.2. Представление графа сети в результате выполнения задания Task\_2*



*Рис.3. Представление графа сети в результате выполнения задания Task\_3*

Из представленных изображений видно, что:

* рис. 1:Узел 0: x = 17, y = 20, цвет = чёрный; Узел 1: x = 19, y = 0, цвет = тёмно-синий; Узел 2: x = 0, y = 2, цвет = тёмно-зелёный; Узел 3: x = 2, y = 4, цвет = тёмно-бирюзовый; Узел 4: x = 4,y = 6, цвет = тёмно-красный. Связанность – последовательная;
* рис. 2: Связанность остаётся как для предыдущего Task. Меняются координаты узлов; Узел 0: x = 1, y = 20; Узел 1: x = 5, y = 10; Узел 2: x = 8, y = 0; Узел 3: x = 12, y = 12; Узел 4: x = 0, y = 2.
* рис. 3: координаты узлов остаются прежними, значения цветов узлов не меняется, узлы 0 и 2 связанны, также как и узлы 3 и 4. Узел 1 не связан ни с одним из других предоставленных узлов.