Промислово-економічний коледж

Національного авіаційного університету

**ЗВІТ**

з лабораторної роботи № 2.11

з дисципліни

«Алгоритми та структури даних»

студента ІІІ денного відділення

групи 427 – ПОМ

Кудін Антон Сергійович

Перевірив викладач:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Рябчук

Оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

м. Київ 2014

**Тема роботи: «Реалізація задач із застосуванням графів»**

**Мета:**

* оволодіти навичками роботи з алгоритмами роботи з графами;
* навчитись вирішувати завдання різними способами.

**Завдання до лабораторної роботи:**

1. Для даних таблиць 1 реалізувати алгоритми роботи з графами.
2. Визначити час виконання задач.

|  |  |
| --- | --- |
| 14 | Знайти найдовший шлях між вершинами X та Y. |

**Розв’язання**

*Код програми:*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Diagnostics;

namespace LAB11\_Graphs

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public static int[,] graph = {

{ 0, 0, 14, 0, 0, 21, 0, 14},

{ 0, 0, 0, 15, 13, 0, 0, 13},

{ 14, 0, 0, 12, 0, 0, 0, 0},

{ 0, 15, 0, 0, 8, 0, 0, 0},

{ 0, 0, 0, 8, 0, 0, 13, 0},

{ 21, 0, 0, 0, 5, 0, 0, 0},

{ 0, 0, 0, 0, 13, 0, 0, 16},

{14, 0, 0, 0, 0, 0, 16, 0}

};

public static int M = 8;

public int[] ways;

bool[] include;

int i,N, start, finish;

List<List<int>> pathes = new List<List<int>>();

private void buttonSearch\_Click(object sender, EventArgs e)

{

Stopwatch SW = new Stopwatch();

SW.Start();

ways = new int[8];

for (i = 0; i < M; i++)

{

ways[i] = 0;

}

include = new bool[M];

for (i = 0; i < N; i++)

{

include[i] = false;

}

start = (int)numericUpDownX.Value - 1;

finish = (int)numericUpDownY.Value - 1;

ways[0] = start;

include[start] = true;

label1.Text = "";

pathes.Clear();

poisk(start, finish, 1);

if (pathes.Count == 0)

{

label1.Text = "Немає шляхів!";

return;

}

else

{

label1.Text = "Всі шляхи:\r\n";

for (int i = 0; i < pathes.Count; i++)

{

for (int j = 0; j < pathes[i].Count; j++)

{

label1.Text += "" + (pathes[i][j] + 1) + " ";

}

label1.Text += "\r\n";

}

if (pathes.Count == 1)

return;

label1.Text += "\r\nНайдовший шлях:\r\n";

int max = 0;

int maxi = 0;

for (int i = 0; i < pathes.Count; i++)

{

if (pathes.ElementAt(i).Count > max)

{

maxi = i;

max = pathes.ElementAt(i).Count;

}

}

for (int i = 0; i < pathes[maxi].Count; i++)

{

label1.Text += "" + (pathes[maxi][i] + 1) + " ";

}

}

SW.Stop();

label2.Text = SW.ElapsedMilliseconds.ToString() + " мс";

}

void poisk(int s, int f, int iskomaya)

{

N = 8;

if (s == f)

{

List<int> list = new List<int>();

for (i = 0; i < iskomaya; i++)

{

label1.Text += "" + (ways[i] + 1) + " ";

list.Add(ways[i]);

}

pathes.Add(list);

}

else

for (int tek = 0; tek < N; tek++)

{

if ((graph[s, tek] > 0) && (!include[tek]))

{

ways[iskomaya] = tek;

include[tek] = true;

poisk(tek, f, iskomaya + 1);

include[tek] = false;

ways[iskomaya] = 0;

}

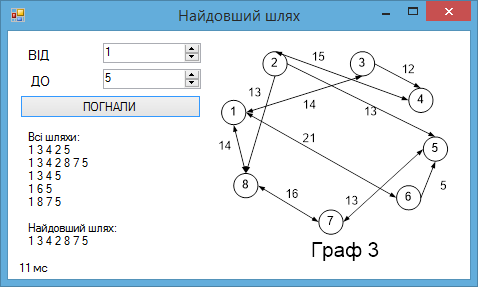
}

}

}

}

*Скріншоти виконанняпрограми:*



***Висновки:*** *Освоїв алгоритми пошуку на графах та застосував їх для знаходження найдовшого шляху між точками.*