Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра “Вычислительная техника”

Пояснительная Записка

К курсовому проектированию

По курсу “Логика и основы алгоритмизации

в инженерных задачах”

на тему “Реализация алгоритма Флойда-Уоршелла поиска

кратчайших путей в графе”

Выполнил:

Студент группы 19ВВ2

Пристяжнюк А. М.

Принял:

Митрохин М.А.

Юрова О.В.

Пенза 2021

**Оглавление**

[Пояснительная Записка 1](#_Toc88674220)

[Реферат 3](#_Toc88674221)

[Введение 4](#_Toc88674222)

[1.Постановка задачи 5](#_Toc88674223)

[2.Теоретическая часть 6](#_Toc88674224)

[3.Описание алгоритма программы 9](#_Toc88674225)

[4.Описание программы 10](#_Toc88674226)

[5.Тестирование программы 16](#_Toc88674227)

[6.Ручной расчет задачи 18](#_Toc88674228)

[Заключение 20](#_Toc88674229)

[Список литературы 21](#_Toc88674230)

[Приложение А. Листинг 22](#_Toc88674231)

Реферат

Отчет 28 страниц, 16 рисунков

Граф, теория графов, Алгоритм Флойда-Уоршелла, алгоритм поиска кратчайших путей

Цель исследования – разработка программы, способная рассчитать кратчайший путь между вершинами графа

В работе используется алгоритм Флойда-Уоршелла, с его помощью можно найти кратчайший путь между вершинами графа

Введение

В математической теории графов и информатике граф — это совокупность непустого множества вершин и множества пар вершин. Объекты представляются как вершины, или узлы графа, а связи — как дуги, или рёбра. Для разных областей применения виды графов могут различаться направленностью, ограничениями на количество связей и дополнительными данными о вершинах или рёбрах.

Многие структуры, представляющие практический интерес в математике и информатике, могут быть представлены графами.

Математические модели в виде графов широко используются при моделировании разнообразных явлений, процессов и систем. Многие теоретические и реальные прикладные задачи могут быть решены при помощи тех или иных процедур анализа графовых моделей. В качестве примера можно привести задачу составления маршрута движения транспорта между различными городами при заданном расстоянии между населенными пунктами.

В данной работе рассматривается алгоритм Флойда — Уоршелла — алгоритм для нахождения кратчайших расстояний между всеми вершинами взвешенного ориентированного графа. Разработан в 1962 году Робертом Флойдом и Стивеном Уоршеллом.

1.Постановка задачи

Требуется разработать программу, которая реализует поиск кратчайших путей во взвешенном графе с помощью алгоритма Флойда-Уоршелла.

Программа должна содержать в себе:

-Текстовое или графическое меню

-Возможность задания пользователем размера графа

-Возможность выбора автоматического или ручного задания графа

-Возможность сохранения результатов работы программы

2.Теоретическая часть

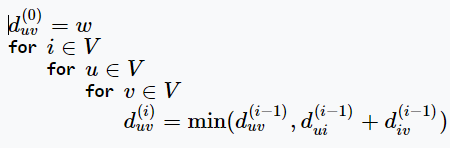
Обозначим длину кратчайшего пути между вершинами *u* и *v*, содержащего, помимо *u* и *v*, только вершины из множества:



На каждом шаге алгоритма, мы будем брать очередную вершину (пусть её номер — i) и для всех пар вершин *u* и *v* вычислять:

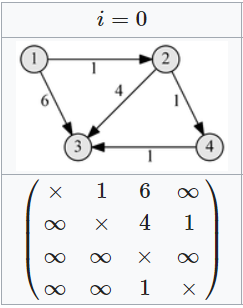
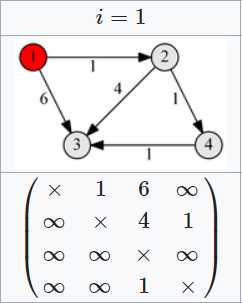


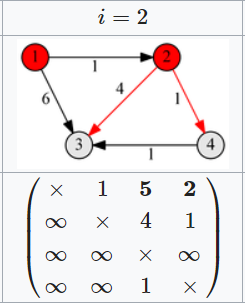
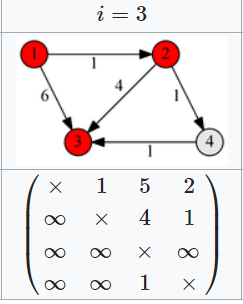
То есть, если кратчайший путь из *u* в *v*, содержащий только вершины из множества {1..i} , проходит через вершину *i*, то кратчайшим путем из *u* в *v* является кратчайший путь из u в *i*, объединенный с кратчайшим путем из *i* в *v*. В противном случае, когда этот путь не содержит вершины *i*, кратчайший путь из *u* в *v*, содержащий только вершины из множества {1..*i*} является кратчайшим путем из *u* в *v*, содержащим только вершины из множества {1..*i*-1}

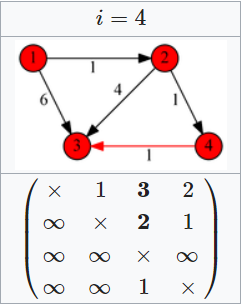


В итоге получаем, что матрица  и является искомой матрицей кратчайших путей, поскольку содержит в себе длины кратчайших путей между всеми парами вершин, имеющих в качестве промежуточных вершин вершины из множества {1..n}, что есть попросту все вершины графа. Такая реализация работает за Θ(n3) времени и использует Θ(n3) памяти

Пример работы алгоритма:



3.Описание алгоритма программы

При заполнении матрицы (ручном или автоматическом) используется динамический двумерный массив. Если между вершинами графа отсутствует прямой путь, то в массив записывается значение =10000.

После того как матрица выводится на экран начинает работать алгоритм Флойда-Уоршелла.

С помощью вложенных циклов производится поиск кратчайших путей. Если соблюдается условие mas[i][j] >mas[i][k] + mas[k][j] ,то в mas[i][j] записывается сумма mas[i][k] + mas[k][j].

После вышеперечисленных манипуляций измененная матрица выводится на экран и пользователю предлагается сохранить результат в файл формата .txt.

Ниже представлен псевдокод для данного алгоритма:

для k=0 пока k<N k=k+1

для i=0 пока i<N i=i+1

для j=0 пока j<N j=j+1

если (mas[i][j] больше mas[i][k] + mas[k][j]

то mas [i][j] = mas[i][k]+mas[k][j]

конец условия

4.Описание программы

Для написания данной программы использован язык программирования С++. Язык программирования С++ - универсальный язык программирования, который завоевал особую популярность у программистов, благодаря сочетанию возможностей языков программирования высокого и низкого уровней.

Проект был создан в виде классического приложения Windows (Visual C++) с использованием форм.

При запуске программы пользователю представляется возможность выбрать количество вершин в матрице, вероятность создания дуги между вершинами и максимальное расстояние между вершинами. Также предоставляется возможность сохранения матрица в файл и открытие матрицы из файла для дальнейшей обработки алгоритмом.

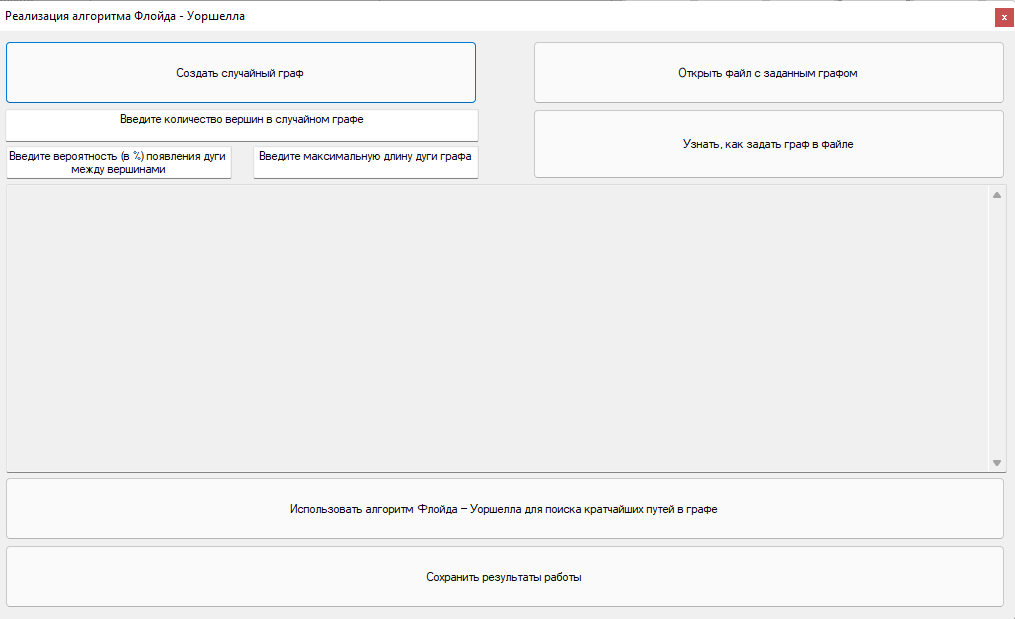


Рисунок 1-меню программы.

Для начала работы программы, пользователь должен ввести значения в поля вверху программы.

Далее происходит инициализация динамического массива и его заполнение псевдослучайными числами. (Если связи между вершинами графа нет, то пишется число -1)

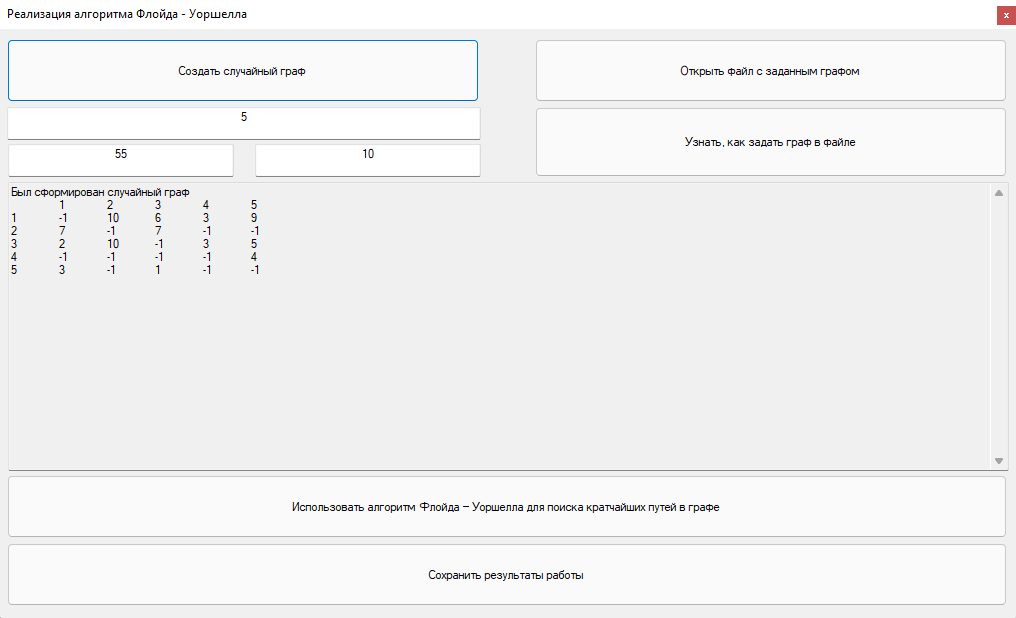


Рисунок 2 - заполнение массива псевдослучайными числами

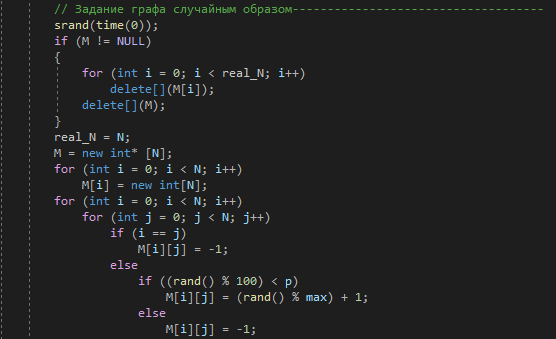


Рисунок 3 - код автоматического заполнения массива

После заполнения производится вывод матрицы на экран.

Затем происходит непосредственная реализация алгоритма Флойда-Уоршелла для поиска кратчайших путей между вершинами взвешенного графа.

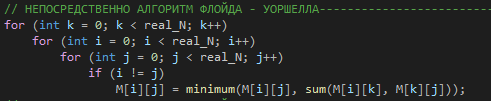


Рисунок 4 - код алгоритма Флойда-Уоршелла

По завершении работы данного алгоритма результат выводится на экран и пользователь может сохранить полученный результат в файл.

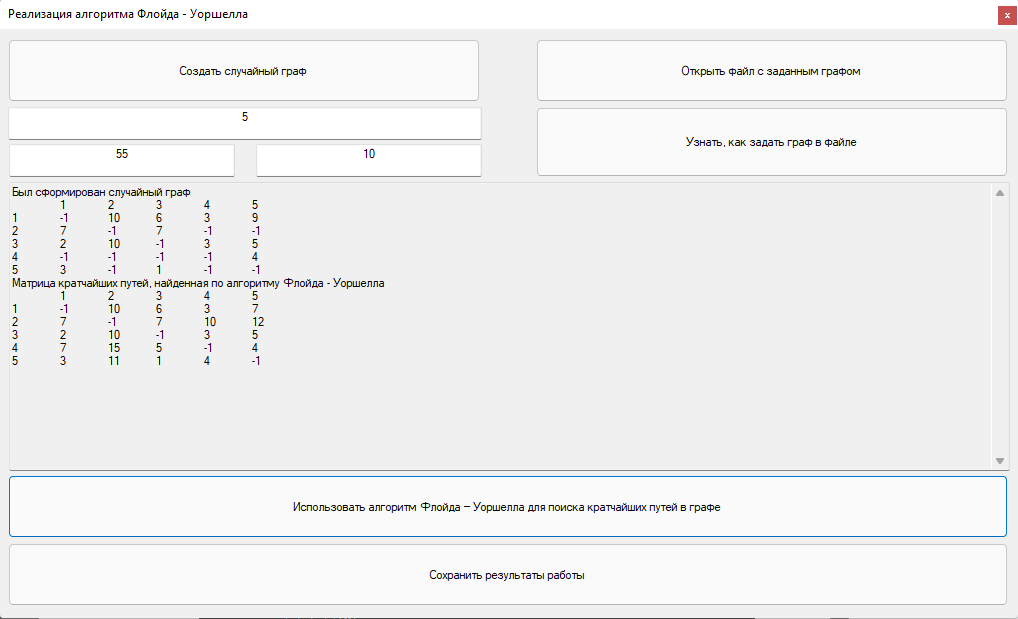


Рисунок 5 - результат работы алгоритма

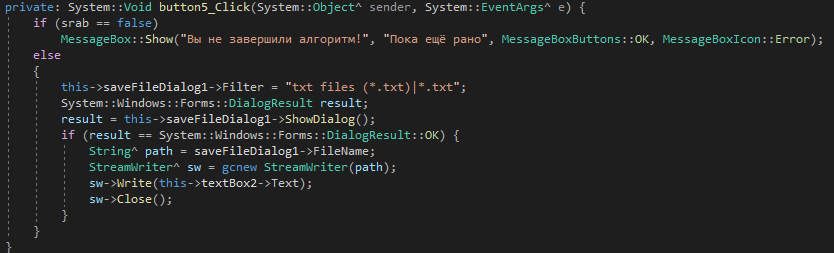


Рисунок 6 - код для записи результата в файл

После нажатия на кнопку сохранения результата работы программы, пользователю дается на выбор директория сохранения и задание имени файла.

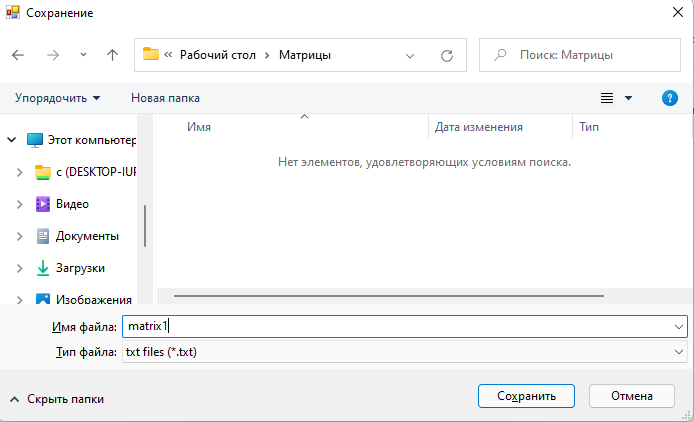


Рисунок 7 – сохранение файла

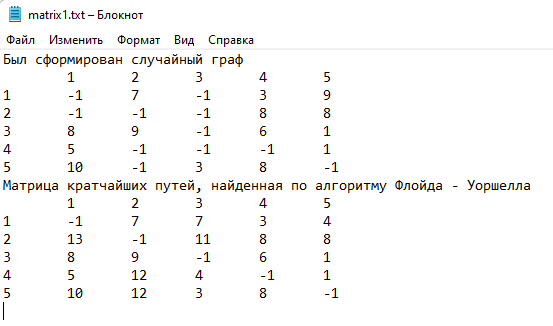


Рисунок 8 - файл с результатом работы программы

Видоизменили файл для открытия его в программе для дальнейшего использования алгоритма, где первая строка – число вершин, а последующие строки являются матрицей с расстояниями между вершинами (если ребра нет пишется -1)

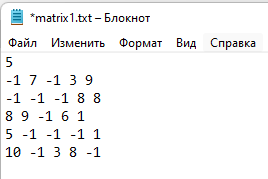


Рисунок 9 – видоизмененный файл

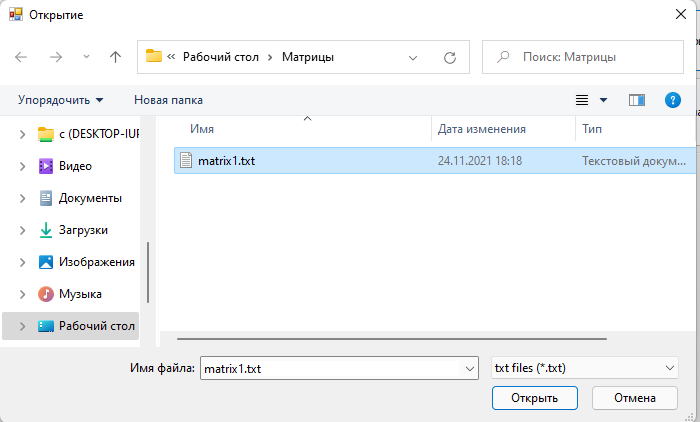


Рисунок 10 – открытие файла

Выведется сообщение, что файл успешно был открыт

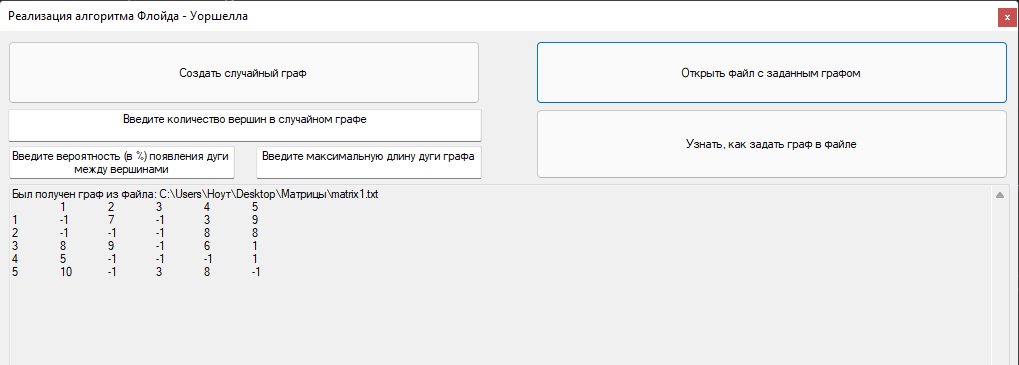


Рисунок 11 – сообщение об успешном прочтении файла

5.Тестирование программы

Среда разработки Microsoft Visual Studio 2010 предоставляет все средства, необходимые при разработке и отладке многомодульной программы.

Тестирование проводилось в рабочем порядке, в процессе разработки,

после завершения написания программы. В ходе тестирования было выявлено и исправлено множество проблем, связанных с вводом данных, изменением дизайна выводимых данных, алгоритмом программы, взаимодействием функций.

Ниже продемонстрирован результат тестирования программы:

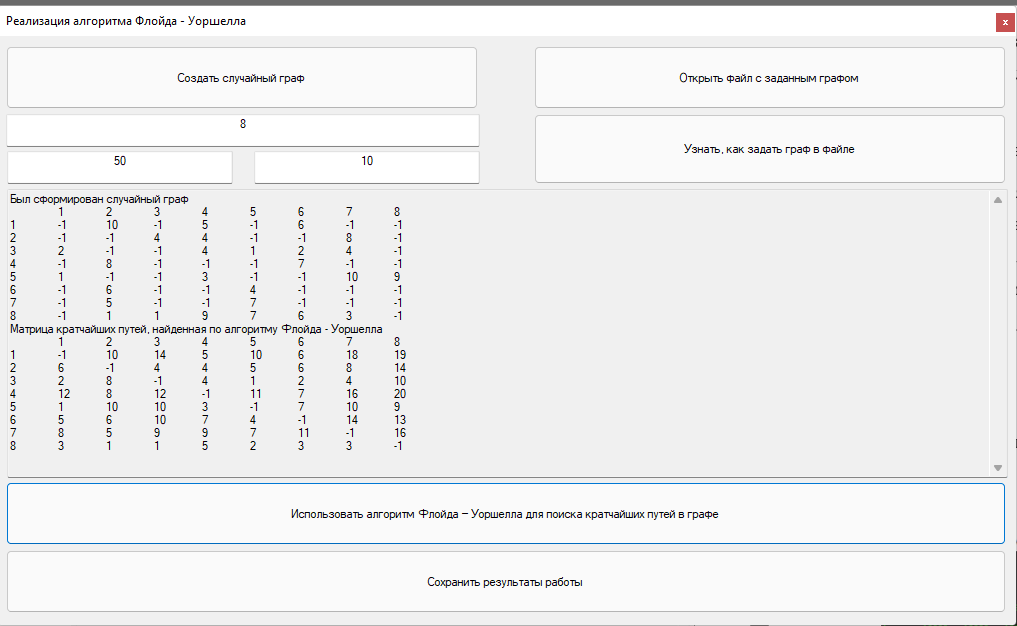


Рисунок 12 - тестирование программы (вершин – 8, вероятность ребра – 50%, макс расстояние - 10)

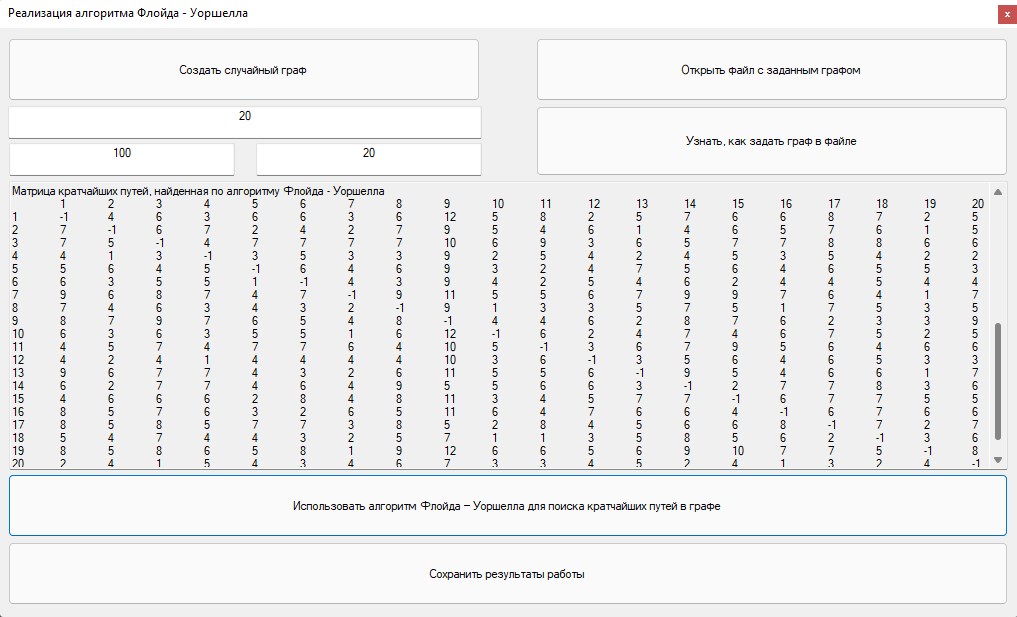


Рисунок 13 - тестирование программы (вершин – 20, вероятность ребра – 100%, макс расстояние - 20)

6.Ручной расчет задачи

Для удобства ручного расчета, возьмем граф с количеством вершин равным 5.

С помощью программы заполним матрицу случайными числами

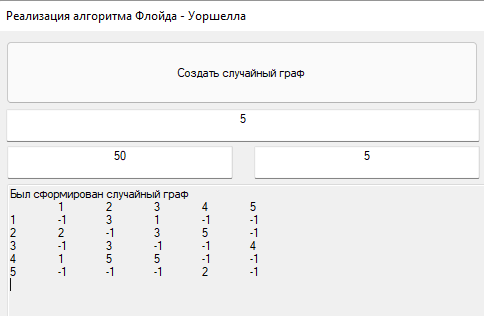


Рисунок 14 - заполнение матрицы

Для наглядности представили матрицу в виде графа

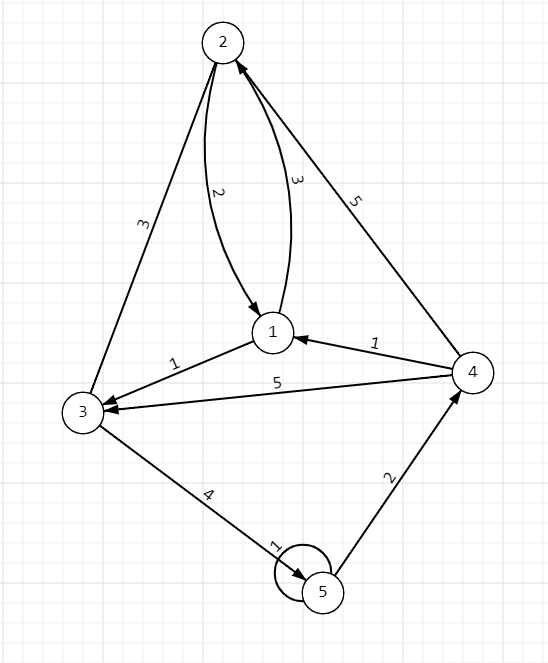


Рисунок 15 - представление графа

Отсюда видно, что вершины 4 и 5 не соединены между собой напрямую. Путем перебора различных вариантов самым коротким оказался следующий путь:

-Из вершины 4 в вершину 1

-Из вершины 1 в вершину 3

-Из вершины 3 в вершину 5

Полученный путь равен 1 + 1 + 4 = 6

Однако, есть еще и другой путь

-Из вершины 4 в вершину 3

-Из вершины 3 в вершину 5

Но такой путь будет уже не 6, а 9, что не является кротчайшим путем.

Сравним результат ручного просчета с результатом работы программы

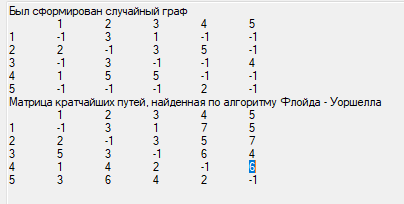


Рисунок 16 – проверка работы программы

Можно сделать вывод, что программа правильно находит кратчайшие пути.

Заключение

Таким образом, в процессе создания данного проекта разработана

программа, реализующая алгоритм Флойда-Уоршелла в Microsoft Visual Studio 2019.

При выполнении данной курсовой работы были получены навыки разработки программ и освоены приемы создания матриц смежностей,

основанных на теории орграфов. Приобретены навыки по осуществлению

алгоритма поиска кратчайших путей. Углублены знания языка программирования C++.

Программа имеет небольшой, но достаточный для использования

функционал возможностей.

Список литературы

* Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн Алгоритмы: построение и анализ — 2-е изд — М.: Издательский дом «Вильямс», 2009. — ISBN 978-5-8459-0857-5.
* Романовский И. В. Дискретный анализ: Учебное пособие для студентов, специализирующихся по прикладной математике и информатике. Изд. 3-е. — СПб.: Невский диалект, 2003. — 320 с. — ISBN 5-7940-0114-3.

Приложение А. Листинг

MyForm.h

#pragma once

#include <ctime>

#include <iostream>

#include <fstream>

#include <msclr\marshal\_cppstd.h>

int\*\* M;

int real\_N;

bool srab = false;

bool matr = false;

int minimum(int x, int y)

{

if (x == -1 && y == -1)

return -1;

if (x == -1)

return y;

if (y == -1)

return x;

if (x < y)

return x;

if (x > y)

return y;

return x;

}

int sum(int x, int y)

{

if (x == -1 || y == -1)

return -1;

return x + y;

}

namespace Project49 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::IO;

/// <summary>

/// Сводка для MyForm1

/// </summary>

public ref class MyForm1 : public System::Windows::Forms::Form

{

public:

MyForm1(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm1()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::OpenFileDialog^ openFileDialog1;

private: System::Windows::Forms::SaveFileDialog^ saveFileDialog1;

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::TextBox^ textBox4;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->openFileDialog1 = (gcnew System::Windows::Forms::OpenFileDialog());

this->saveFileDialog1 = (gcnew System::Windows::Forms::SaveFileDialog());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->textBox4 = (gcnew System::Windows::Forms::TextBox());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(8, 10);

this->button1->Margin = System::Windows::Forms::Padding(2);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(472, 63);

this->button1->TabIndex = 0;

this->button1->Text = L"Создать случайный граф";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm1::button1\_Click);

//

// button2

//

this->button2->Location = System::Drawing::Point(536, 10);

this->button2->Margin = System::Windows::Forms::Padding(2);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(472, 63);

this->button2->TabIndex = 1;

this->button2->Text = L"Открыть файл с заданным графом";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &MyForm1::button2\_Click);

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(8, 78);

this->textBox1->Margin = System::Windows::Forms::Padding(2);

this->textBox1->Multiline = true;

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(474, 33);

this->textBox1->TabIndex = 2;

this->textBox1->Text = L"Введите количество вершин в случайном графе";

this->textBox1->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;

this->textBox1->Click += gcnew System::EventHandler(this, &MyForm1::textBox1\_Click);

this->textBox1->TextChanged += gcnew System::EventHandler(this, &MyForm1::textBox1\_TextChanged);

//

// button3

//

this->button3->Location = System::Drawing::Point(536, 78);

this->button3->Margin = System::Windows::Forms::Padding(2);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(472, 70);

this->button3->TabIndex = 3;

this->button3->Text = L"Узнать, как задать граф в файле";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &MyForm1::button3\_Click);

//

// button4

//

this->button4->Location = System::Drawing::Point(8, 446);

this->button4->Margin = System::Windows::Forms::Padding(2);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(1000, 63);

this->button4->TabIndex = 5;

this->button4->Text = L"Использовать алгоритм Флойда — Уоршелла для поиска кратчайших путей в графе";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &MyForm1::button4\_Click);

//

// button5

//

this->button5->Location = System::Drawing::Point(8, 514);

this->button5->Margin = System::Windows::Forms::Padding(2);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(1000, 63);

this->button5->TabIndex = 6;

this->button5->Text = L"Сохранить результаты работы";

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew System::EventHandler(this, &MyForm1::button5\_Click);

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(9, 153);

this->textBox2->Margin = System::Windows::Forms::Padding(2);

this->textBox2->Multiline = true;

this->textBox2->Name = L"textBox2";

this->textBox2->ReadOnly = true;

this->textBox2->ScrollBars = System::Windows::Forms::ScrollBars::Both;

this->textBox2->Size = System::Drawing::Size(999, 289);

this->textBox2->TabIndex = 7;

//

// openFileDialog1

//

this->openFileDialog1->FileName = L"openFileDialog1";

//

// textBox3

//

this->textBox3->Location = System::Drawing::Point(256, 115);

this->textBox3->Margin = System::Windows::Forms::Padding(2);

this->textBox3->Multiline = true;

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(226, 33);

this->textBox3->TabIndex = 8;

this->textBox3->Text = L"Введите максимальную длину дуги графа";

this->textBox3->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;

this->textBox3->Click += gcnew System::EventHandler(this, &MyForm1::textBox3\_Click);

//

// textBox4

//

this->textBox4->Location = System::Drawing::Point(9, 115);

this->textBox4->Margin = System::Windows::Forms::Padding(2);

this->textBox4->Multiline = true;

this->textBox4->Name = L"textBox4";

this->textBox4->Size = System::Drawing::Size(226, 33);

this->textBox4->TabIndex = 9;

this->textBox4->Text = L"Введите вероятность (в %) появления дуги между вершинами";

this->textBox4->TextAlign = System::Windows::Forms::HorizontalAlignment::Center;

this->textBox4->Click += gcnew System::EventHandler(this, &MyForm1::textBox4\_Click);

//

// MyForm1

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->AutoSizeMode = System::Windows::Forms::AutoSizeMode::GrowAndShrink;

this->BackColor = System::Drawing::SystemColors::ButtonFace;

this->ClientSize = System::Drawing::Size(1018, 589);

this->Controls->Add(this->textBox4);

this->Controls->Add(this->textBox3);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->button5);

this->Controls->Add(this->button4);

this->Controls->Add(this->button3);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->button2);

this->Controls->Add(this->button1);

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedToolWindow;

this->Margin = System::Windows::Forms::Padding(2);

this->MaximumSize = System::Drawing::Size(1920, 1080);

this->Name = L"MyForm1";

this->Text = L"Реализация алгоритма Флойда - Уоршелла";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

MessageBox::Show("Формат входного файла: в первой строке находится число N - количество вершин, а далее следует матрица N на N, в которой отображены длины дуг между вершинами. В строке между значениями может стоять пробел или табуляция. Если между вершинами нет дуги, то необходимо поставить значение -1. Диагональные элементы также равны -1. Тип файла: текстовый файл (.txt)", "Конфигурация входного файла", MessageBoxButtons::OK, MessageBoxIcon::Asterisk);

}

private: System::Void textBox1\_Click(System::Object^ sender, System::EventArgs^ e) {

this->textBox1->Text = "";

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

int N = Convert::ToInt32(this->textBox1->Text);

int max = Convert::ToInt32(this->textBox3->Text);

int p = Convert::ToInt32(this->textBox4->Text);

// Задание графа случайным образом------------------------------------

srand(time(0));

if (M != NULL)

{

for (int i = 0; i < real\_N; i++)

delete[](M[i]);

delete[](M);

}

real\_N = N;

M = new int\* [N];

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

M[i] = new int[N];

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

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

if (i == j)

M[i][j] = -1;

else

if ((rand() % 100) < p)

M[i][j] = (rand() % max) + 1;

else

M[i][j] = -1;

// Задание графа случайным образом------------------------------------

// Вывод графа на экран------------------------------------

this->textBox2->Text = "";

this->textBox2->Text += "Был сформирован случайный граф\r\n\t";

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

this->textBox2->Text += i + 1 + "\t";

this->textBox2->Text += "\r\n";

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

{

this->textBox2->Text += i + 1 + "\t";

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

this->textBox2->Text += Convert::ToString(M[i][j]) + "\t";

this->textBox2->Text += "\r\n";

}

// Вывод графа на экран------------------------------------

matr = true;

srab = false;

}

private: System::Void textBox3\_Click(System::Object^ sender, System::EventArgs^ e) {

this->textBox3->Text = "";

}

private: System::Void textBox4\_Click(System::Object^ sender, System::EventArgs^ e) {

this->textBox4->Text = "";

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

// Загрузка графа из файла

this->openFileDialog1->Filter = "txt files (\*.txt)|\*.txt";

System::Windows::Forms::DialogResult result;

result = this->openFileDialog1->ShowDialog();

if (result == System::Windows::Forms::DialogResult::OK) {

String^ path = openFileDialog1->FileName;

msclr::interop::marshal\_context context;

std::string PATH = context.marshal\_as<std::string>(path);

std::ifstream file(PATH);

int N;

file >> N;

if (M != NULL)

{

for (int i = 0; i < real\_N; i++)

delete[](M[i]);

delete[](M);

}

real\_N = N;

M = new int\* [N];

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

M[i] = new int[N];

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

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

file >> M[i][j];

// Вывод графа на экран------------------------------------

this->textBox2->Text = "";

this->textBox2->Text += "Был получен граф из файла: " + path + " \r\n\t";

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

this->textBox2->Text += i + 1 + "\t";

this->textBox2->Text += "\r\n";

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

{

this->textBox2->Text += i + 1 + "\t";

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

this->textBox2->Text += Convert::ToString(M[i][j]) + "\t";

this->textBox2->Text += "\r\n";

}

// Вывод графа на экран------------------------------------

}

matr = true;

srab = false;

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

if (matr)

{

// НЕПОСРЕДСТВЕННО АЛГОРИТМ ФЛОЙДА - УОРШЕЛЛА----------------------------------

for (int k = 0; k < real\_N; k++)

for (int i = 0; i < real\_N; i++)

for (int j = 0; j < real\_N; j++)

if (i != j)

M[i][j] = minimum(M[i][j], sum(M[i][k], M[k][j]));

// НЕПОСРЕДСТВЕННО АЛГОРИТМ ФЛОЙДА - УОРШЕЛЛА----------------------------------

// Вывод кратчайших путей на экран------------------------------------

this->textBox2->Text += "Матрица кратчайших путей, найденная по алгоритму Флойда - Уоршелла\r\n\t";

for (int i = 0; i < real\_N; i++)

this->textBox2->Text += i + 1 + "\t";

this->textBox2->Text += "\r\n";

for (int i = 0; i < real\_N; i++)

{

this->textBox2->Text += i + 1 + "\t";

for (int j = 0; j < real\_N; j++)

this->textBox2->Text += Convert::ToString(M[i][j]) + "\t";

this->textBox2->Text += "\r\n";

}

// Вывод кратчайших путей на экран------------------------------------

srab = true;

}

else

MessageBox::Show("Вы не задали граф!", "Граф не задан", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e) {

if (srab == false)

MessageBox::Show("Вы не завершили алгоритм!", "Пока ещё рано", MessageBoxButtons::OK, MessageBoxIcon::Error);

else

{

this->saveFileDialog1->Filter = "txt files (\*.txt)|\*.txt";

System::Windows::Forms::DialogResult result;

result = this->saveFileDialog1->ShowDialog();

if (result == System::Windows::Forms::DialogResult::OK) {

String^ path = saveFileDialog1->FileName;

StreamWriter^ sw = gcnew StreamWriter(path);

sw->Write(this->textBox2->Text);

sw->Close();

}

}

}

private: System::Void textBox1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

}

};

}

MyForm.cpp

#include "MyForm.h"

#include <Windows.h>

using namespace Project49;

[STAThreadAttribute]

int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Application::Run(gcnew MyForm1);

return 0;

}