Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно-графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Дискретная математика*** |
|  |  |
| на тему | Реализация алгоритма поиска кратчайшего пути |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020-РГР-02.03.02-№ 3-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Дудикова Дмитрия Евгеньевича | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | ФИТ-232 | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | *Фундаментальная информатика и информационные технологии* | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | | 31.05.2024 | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | | | |
|  |  |  |  | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

Омск 2024

**Содержа**ние

[Введение 3](#_Toc22848)

Постановка задачи 4

[Теоретическая часть 5](#_Toc32058)

[Разработка кода 6](#_Toc16304)

[Результаты 8](#_Toc2637)

[Заключение 9](#_Toc23786)

[Список используемой литературы 1](#_Toc21612)0

# Введение

Дискретная математика играет важную роль в развитии и применении современных алгоритмов программирования. Её методы и теории находят широкое применение в разработке программного обеспечения, обеспечивая эффективность и надёжность вычислительных процессов. В этой расчётно-графической работе будет рассматриваться основной алгоритм дискретной математики, алгоритм Флойда-Уоршелла, который позволяет найти кратчайший путь.

В данной работе этот алгоритм реализован на языке программирования C#.

Целью расчётно-графической работы является демонстрация применения алгоритмов дискретной математики для решения практических задач в программировании, что подчёркивает их значимость и универсальность в области информационных технологий.

# Постановка задачи

Имеется некоторое количество авиарейсов между городами мира, для каждого известна стоимость. Стоимость перелёта из A в B может быть не равна стоимости перелёта из B в A. Найти маршруты минимальной стоимости между городами (возможно, с пересадками).

**Формат входных данных**

Во входном файле записано сначала число N (1<=N<=100), определявшее количество рассматриваемых городов. Затем идет число M - количество авиарейсов между городами мира, далее идет описание самих дорог. Каждое сообщение между городами задается тремя числами - номерами городов и стоимостью перелета.

**Формат выходных данных**

На экран выведите числа - суммарные стоимости маршрутов с указанием номеров городов или -1, если добраться невозможно.

# Теоретическая часть

Алгоритм Флойда-Уоршелла используется для нахождения кратчайших путей между всеми парами вершин взвешенного ориентированного или неориентированного графа. Он работает с матрицей расстояний между парами вершин и обновляет эту матрицу, чтобы получить минимальные расстояния.

Пусть дан граф с n вершинами. Матрица расстояний размера инициализируется так, что равно весу ребра из вершины в вершину , если такое ребро существует, и бесконечности (или большому числу) в противном случае.

Алгоритм Флойда-Уоршелла выполняет следующие шаги:

1. **Инициализация**: Пусть — исходная матрица расстояний графа.
2. **Шаги итерации**: Для каждой вершины от 1 до , для каждой пары вершин i, j (где i, j могут быть равными), обновляем расстояние:

Это означает, что для каждой пары вершин , мы проверяем, является ли текущее известное расстояние между i и j меньше, чем расстояние через вершину . Если расстояние через короче, мы обновляем соответственно.

1. **Результат**: После nnn шагов матрица будет содержать минимальные расстояния между всеми парами вершин.

# Разработка кода

Этот код реализует реализует программу на C#, использующую алгоритм Флойда-Уоршелла для нахождения кратчайших путей в графе, описывающем сеть авиаперелетов между городами с известными стоимостями перелетов. Программа предлагает пользователю несколько опций в главном меню: вычисление, об авторе программы, описание постановки задачи и выход. Пользователь выбирает пункт меню, после чего выполняется соответствующая функция.

using System;

using System.Collections.Generic;

using System.IO;

class Program

{

const int INF = int.MaxValue / 2;

static void Main()

{

// Путь к входному файлу

string inputFilePath = @"E:\Temp\ConsoleApp1\input.txt";

string outputFilePath = @"E:\Temp\ConsoleApp1\output.txt";

// Главное меню

while (true)

{

Console.WriteLine("Главное меню");

Console.WriteLine("1. Вычисление");

Console.WriteLine("2. Об авторе программы");

Console.WriteLine("3. Описание постановки задачи");

Console.WriteLine("4. Выход");

Console.Write("Выберите пункт меню: ");

string choice = Console.ReadLine();

switch (choice)

{

case "1":

PerformCalculation(inputFilePath, outputFilePath);

Results();

break;

case "2":

DisplayAuthorInfo();

break;

case "3":

DisplayTaskDescription();

break;

case "4":

return;

default:

Console.WriteLine("Некорректный выбор. Попробуйте снова.");

break;

}

}

}

static void PerformCalculation(string inputFilePath, string outputFilePath)

{

// Чтение входных данных из файла

string[] lines = File.ReadAllLines(inputFilePath);

int index = 0;

int N = int.Parse(lines[index++]);

int M = int.Parse(lines[index++]);

int[,] dist = new int[N, N];

int[,] nextNode = new int[N, N];

// Инициализация матриц

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

{

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

{

if (i == j)

{

dist[i, j] = 0;

}

else

{

dist[i, j] = INF;

}

nextNode[i, j] = -1;

}

}

// Заполнение исходных данных

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

{

var line = lines[index++].Split();

int u = int.Parse(line[0]) - 1;

int v = int.Parse(line[1]) - 1;

int cost = int.Parse(line[2]);

if (u < 0 || u >= N || v < 0 || v >= N)

{

throw new ArgumentOutOfRangeException("Номера городов должны быть в пределах от 1 до N.");

}

dist[u, v] = cost;

nextNode[u, v] = v;

}

// Алгоритм Флойда-Уоршелла

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

{

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

{

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

{

if (dist[i, k] < INF && dist[k, j] < INF && dist[i, j] > dist[i, k] + dist[k, j])

{

dist[i, j] = dist[i, k] + dist[k, j];

nextNode[i, j] = nextNode[i, k];

}

}

}

}

// Сохранение результатов в файл

using (StreamWriter writer = new StreamWriter(outputFilePath))

{

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

{

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

{

if (i != j)

{

if (dist[i, j] == INF)

{

writer.WriteLine($"{i + 1} {j + 1} -1");

}

else

{

var path = ConstructPath(nextNode, i, j);

string pathStr = string.Join(" -> ", path);

writer.WriteLine($"{i + 1} {j + 1} {dist[i, j]} {pathStr}");

}

}

}

}

}

Console.WriteLine("Вычисление завершено. Результаты сохранены в output.txt.");

}

static List<int> ConstructPath(int[,] nextNode, int u, int v)

{

if (nextNode[u, v] == -1)

return new List<int>();

List<int> path = new List<int> { u + 1 };

while (u != v)

{

u = nextNode[u, v];

path.Add(u + 1);

}

return path;

}

static void DisplayAuthorInfo()

{

Console.WriteLine("Программа разработана студентом первого курса ОмГТУ группы ФИТ-232 Дудиковым Дмитрием Евгеньевичем");

}

static void DisplayTaskDescription()

{

Console.WriteLine("Описание постановки задачи:");

Console.WriteLine("Дана сеть авиаперелетов между городами с известной стоимостью перелета.");

Console.WriteLine("Необходимо найти маршруты минимальной стоимости между всеми парами городов, учитывая возможные пересадки.");

}

static void Results()

{

string outputFilePath = @"E:\Temp\ConsoleApp1\output.txt";

while (true)

{

Console.WriteLine("Введите номера городов (n и j) для получения маршрута или 'exit' для выхода:");

string input = Console.ReadLine();

if (input.ToLower() == "exit")

break;

var parts = input.Split();

if (parts.Length != 2 || !int.TryParse(parts[0], out int n) || !int.TryParse(parts[1], out int j))

{

Console.WriteLine("Некорректный ввод. Попробуйте снова.");

continue;

}

string[] lines = File.ReadAllLines(outputFilePath);

bool routeFound = false;

foreach (var line in lines)

{

var segments = line.Split();

int fromCity = int.Parse(segments[0]);

int toCity = int.Parse(segments[1]);

int cost = int.Parse(segments[2]);

if (fromCity == n && toCity == j)

{

if (cost == -1)

{

Console.WriteLine($"Маршрут из города {n} в город {j} невозможен.");

}

else

{

string path = string.Join(" ", segments, 3, segments.Length - 3);

Console.WriteLine($"Маршрут из города {n} в город {j}: {cost}, Путь: {path}");

}

routeFound = true;

break;

}

}

if (!routeFound)

{

Console.WriteLine("Маршрут не найден.");

}

}

}

}

# Результаты

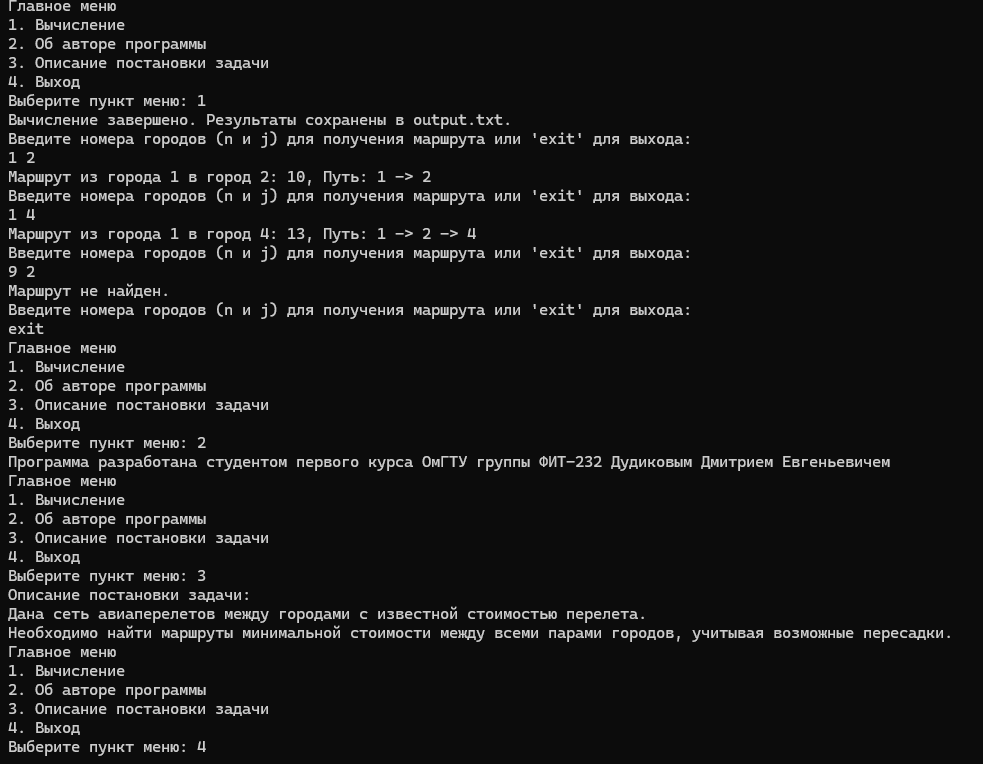


Рисунок 1 – Результат работы программы

# Заключение

В данной расчётно-графической работе был исследован алгоритм Флойда-Уоршелла, применяемый для нахождения кратчайшего пути во взвешенном графе от одной вершины до всех остальных. Алгоритм показал эффективность и высокую точность в определении оптимального маршрута, основываясь на минимальной стоимости прохождения через вершины.

С помощью языка программирования C# была разработана программа, реализующая алгоритм Флойда-Уоршелла.

# Список используемой литературы

1. Алгоритм Флойда-Уоршелла // Википедия URL: https://ru.wikipedia.org/wiki/Алгоритм\_Флойда\_—\_Уоршелла (дата обращения: 29.05.2024).