ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчет по лабораторной работе №1**

по курсу «Дискретная математика»

**«Связность графов»**

Выполнил: студент группы РИС-23-3Б

Асташин Д. А.

Проверил: старший преподаватель кафедры ИТАС

Рустамханова Г.И.

2024 г.

**Цель работы**

Разработка программы для анализа связности неориентированного графа с использованием матрицы смежности.

**Постановка задачи**

Граф представляет собой матрицу размером 10×10 из единиц и нулей, разделенных пробелом.

Реализовать:

1. Считывание бинарной матрицы из текстового файла;
2. Вывод матрицы на экран;
3. Обход в глубину;
4. Вывод данных: количество компонент связности и вершины в каждой компоненте;
5. Матрицу достижимости.

**Реализация**

1. Чтение матрицы из файла:

Матрица считывается построчно с использованием File.ReadAllLines. Каждая строка разбивается на элементы с помощью метода Split (рисунок №1). Данные записываются в двумерный массив.

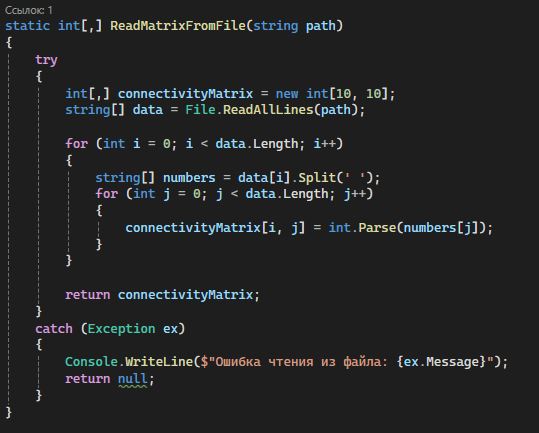


Рис.1 – чтение матрицы из файла

1. Обход матрицы:

Для нахождения компонент связности используется обход в глубину, реализованный в методе DFS, показанный на рисунке №3.

Работа алгоритма:

1. Создаём стек и кладём в него стартовую вершину.
2. Пока стек не пуст, достаём вершину v:
   * Если она не посещена, отмечаем её как посещённую.
   * Добавляем в список текущей компоненты связности.
   * Проверяем всех соседей вершины v: если сосед не посещён, добавляем его в стек.
3. Повторяем, пока стек не опустеет.

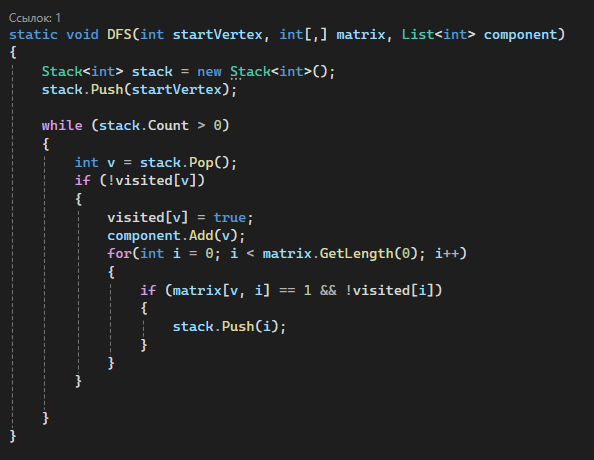


Рис.3 – метод DFS

1. Подсчет количества компонент связности

Метод CountConnectivityComponents(int[,] matrix), представленный на рисунке №4, предназначен для определения количества связных компонент.

Метод использует **обход в глубину (DFS)** для поиска компонент связности. Он проходит по всем вершинам графа, и, если вершина **не была посещена**, это означает, что началась новая компонента связности. В этом случае вызывается **DFS**, который находит все вершины, принадлежащие данной компоненте. Найденная компонента добавляется в общий список компонент.

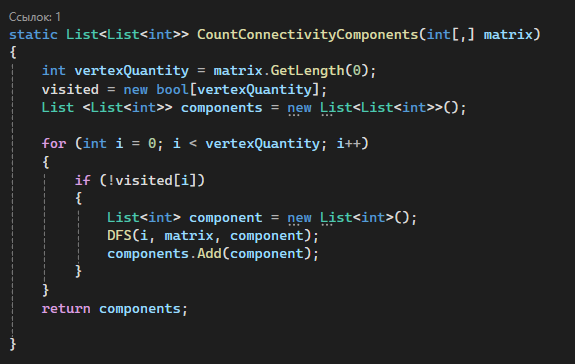


Рис. 4 - Метод CountConnectivityComponents(int[,] matrix)

1. Матрица достижимости

Для нахождения матрицы достижимости используется алгоритм Уоршалла, реализованный в методе WarshallAlgorithm(int[,] matrix) (Рисунок №5).

Принцип работы:

1. Инициализация: Входными данными служит матрица смежности matrix, которая отражает существование рёбер между вершинами (1 — есть ребро, 0 — нет).
2. Копирование матрицы: Создаётся копия reachabilityMatrix, которая будет обновляться в ходе работы алгоритма.
3. Промежуточные вершины:
   * Перебираем вершины k, которые могут быть промежуточными точками на пути между i и j.
   * Для каждой пары i, j, если существует путь i → k и k → j, то существует путь i → j, поэтому reachabilityMatrix[i, j] становится равной 1.
4. Результат: После выполнения алгоритма reachabilityMatrix содержит 1 в тех ячейках, где есть путь между вершинами, и 0, если пути нет.

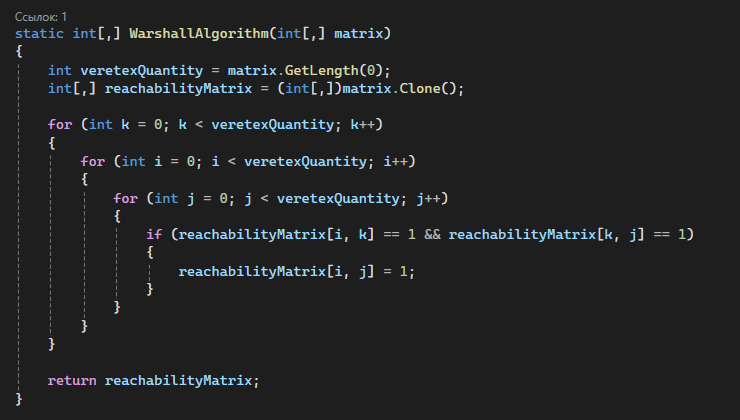
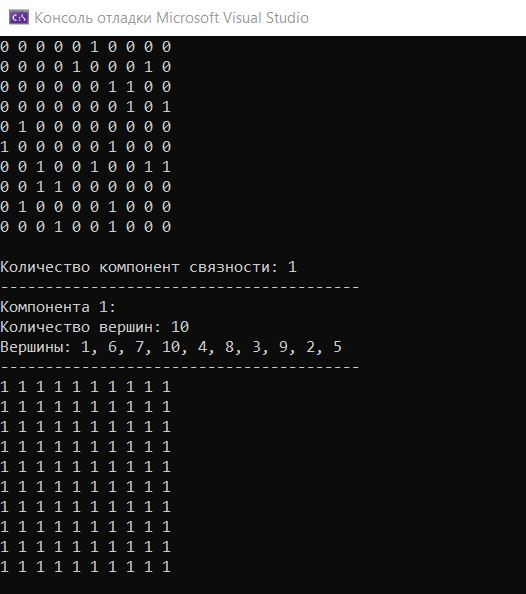


Рис. 5 – Алгоритм Уоршалла

**Результат работы программы**

****

**Заключение**

В результате была разработана программа, определяющая количество компонент связности графа, число вершин в каждой компоненте и их состав.  
В процессе работы были изучены алгоритм обхода графа в глубину (DFS) для поиска компонент связности и алгоритм Уоршалла для построения матрицы достижимости.

[GitHub](https://github.com/astidii/PNRPU/tree/main/Discrete%20mathematics/Graph%20Connectivity)