Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

Дискретная математика

Семестр 4

Лабораторная работа №5

Выполнил  
студент группы РИС-22-2б  
Зубов Р.А.

Проверила  
старший преподаватель кафедры ИТАС  
Рустамханова Г.И.

Пермь, 2024

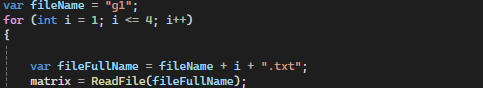
Постановка задачи

На вход поступает матрица смежности без петель (10х10). Ввод матрицы должен быть из файла. Требуется:

1. Построить матрицу достижимости
2. Вычислить компоненты связности

Алгоритмы работы программы

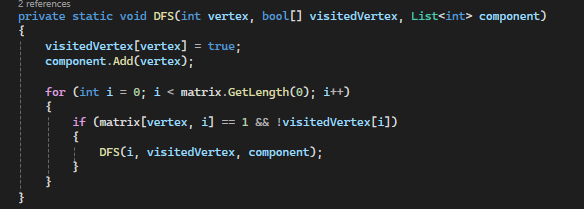
Матрица поступает из файла:



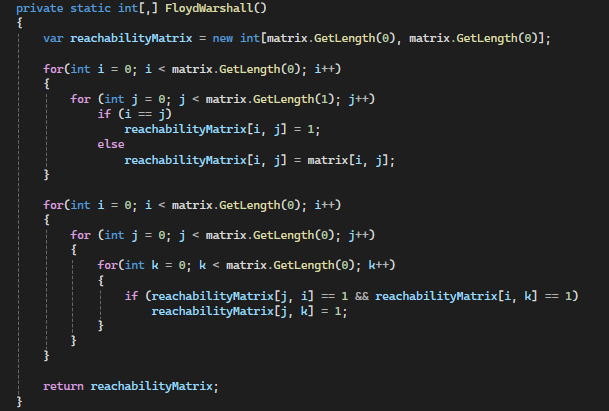
*Рисунок 1 – Ввод матрицы*

Следующий шаг: вывод матрицы из файла. Так как данные из файла поступают в виде строки, следует преобразовать строку в двумерный массив.

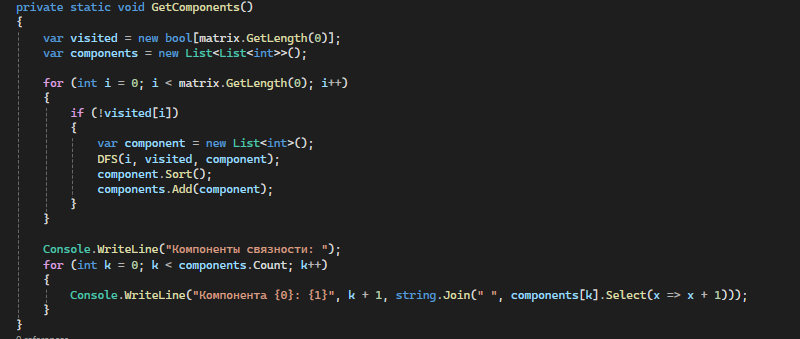
Далее реализованы 2 функции: *FloydWarshall()* и *getComponents()*. Первая функция создает матрицу достижимости по матрице смежности, вторая - вычисляет компоненты связности.



*Рисунок 3 – DFS*

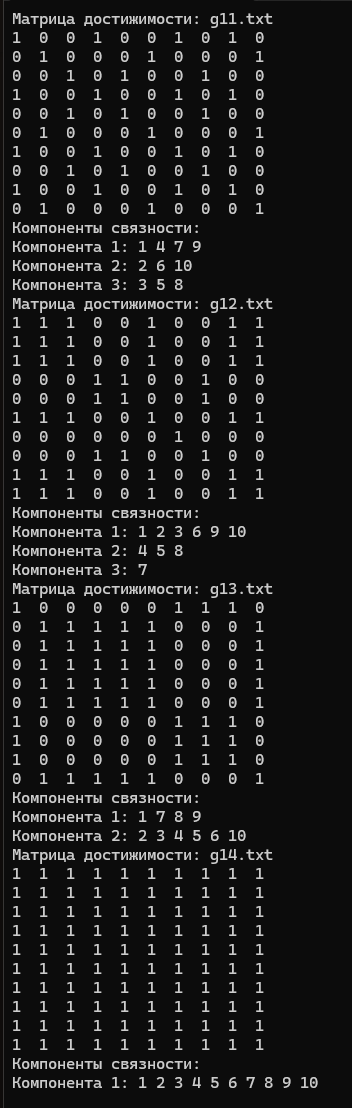


*Рисунок 3 – Матрица достижимости*



*Рисунок 4 – Вычисление компонент связности*

**Результаты работы программы:**

****

**Листинг:**

using System.Collections.Generic;

namespace Lab5

{

public class Program

{

private static int[,] matrix;

private static int[,] ReadFile(string filename)

{

int[,] matrix = new int[10, 10];

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

for (int i = 0; i < 10; i++)

{

string[] values = lines[i].Split(' ');

for (int j = 0; j < 10; j++)

{

matrix[i, j] = int.Parse(values[j]);

}

}

return matrix;

}

private static void PrintMatrix(int[,] matrix)

{

for(int i = 0; i < matrix.GetLength(0); i++)

{

for(int j = 0; j < matrix.GetLength(1); j++)

{

Console.Write(matrix[i, j] + " ");

}

Console.WriteLine();

}

}

private static void DFS(int vertex, bool[] visitedVertex, List<int> component)

{

visitedVertex[vertex] = true;

component.Add(vertex);

for (int i = 0; i < matrix.GetLength(0); i++)

{

if (matrix[vertex, i] == 1 && !visitedVertex[i])

{

DFS(i, visitedVertex, component);

}

}

}

private static int[,] FloydWarshall()

{

var reachabilityMatrix = new int[matrix.GetLength(0), matrix.GetLength(0)];

for(int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

if (i == j)

reachabilityMatrix[i, j] = 1;

else

reachabilityMatrix[i, j] = matrix[i, j];

}

for(int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(0); j++)

{

for(int k = 0; k < matrix.GetLength(0); k++)

{

if (reachabilityMatrix[j, i] == 1 && reachabilityMatrix[i, k] == 1)

reachabilityMatrix[j, k] = 1;

}

}

}

return reachabilityMatrix;

}

private static void GetComponents()

{

var visited = new bool[matrix.GetLength(0)];

var components = new List<List<int>>();

for (int i = 0; i < matrix.GetLength(0); i++)

{

if (!visited[i])

{

var component = new List<int>();

DFS(i, visited, component);

component.Sort();

components.Add(component);

}

}

Console.WriteLine("Компоненты связности: ");

for (int k = 0; k < components.Count; k++)

{

Console.WriteLine("Компонента {0}: {1}", k + 1, string.Join(" ", components[k].Select(x => x + 1)));

}

}

private static void Main(string[] args)

{

var fileName = "g1";

for (int i = 1; i <= 4; i++)

{

var fileFullName = fileName + i + ".txt";

matrix = ReadFile(fileFullName);

Console.WriteLine("Матрица достижимости: " + fileFullName);

var matrixReachability = FloydWarshall();

PrintMatrix(matrixReachability);

GetComponents();

}

}

}

}