Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная кафедра»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Определение характеристик графов»

Выполнили:

студент группы 23ВВВ2

Кокарев Д.С.

Приняли:

Юрова О.В.

Митрохин М.А.

Пенза 2024

**Название**

Определение характеристик графов.

**Цель работы**

Научиться определять характеристика графа.

**Лабораторное задание**

### *Задание 1*

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа G. Выведите матрицу на экран.
2. Определите размер графа G, используя матрицу смежности графа.
3. Найдите изолированные, концевые и доминирующие вершины.

### *Задание 2\**

1. Постройте для графа G матрицу инцидентности.
2. Определите размер графа G, используя матрицу инцидентности графа.
3. Найдите изолированные, концевые и доминирующие вершины.

**Описание метода решения задачи**

*Задание 1*

Была сгенерирована (используя генератор случайных чисел) матрица смежности для неориентированного графа. Главная диагональ матрицы заполнена нулями. Так как граф неориентированный – матрица симметрична относительно главной диагонали. Определен размер графа путём подсчета «1» выше главной диагонали. Также найдены изолированные, концевые и доминирующие вершины, рассматривая каждую строку матрицы смежности поочередно.

*Задание 2*

Был построен для графа матрица инцидентности. Определен размер графа, используя матрицу инцидентности. Найдены изолированные, концевые и доминирующие вершины, рассматривая каждую строку матрицы смежности поочередно.

**Листинг**

**lab5.cpp**

#include <iostream>

#include <ctime>

#include <locale>

int main() {

srand(time(NULL));

setlocale(LC\_ALL, "");

int size;

std::cout << "Введите размер матрицы смежности: ";

std::cin >> size;

if (size <= 0) {

std::cout << "Размер матрицы должен быть > 0" << std::endl;

return 0;

}

std::cout << std::endl;

int\*\* arr = new int\* [size];

for (int i = 0; i < size; ++i) {

arr[i] = new int[size];

}

int count\_smez = 0;

for (int i = 0; i < size; ++i) {

for (int j = 0; j < size; ++j) {

if (i == j) {

arr[i][j] = 0;

}

else if (i > j) {

arr[i][j] = arr[j][i];

}

else {

arr[i][j] = rand() % 2;

if (arr[i][j] == 1) {

count\_smez++;

}

}

}

}

for (int i = 0; i < size; ++i) {

for (int j = 0; j < size; ++j) {

std::cout << arr[i][j] << "\t";

}

std::cout << std::endl;

}

std::cout << "\nКоличество рёбер (размер) графа через матрицу смежности: " << count\_smez << "\n\n";

for (int i = 0; i < size; ++i) {

int g = 0;

for (int j = 0; j < size; ++j) {

if (arr[i][j] == 1) {

g++;

}

}

if (g == 0) std::cout << "Вершина " << i + 1 << " - изолированная" << std::endl;

if (g == 1) std::cout << "Вершина " << i + 1 << " - концевая" << std::endl;

if (g == size - 1) std::cout << "Вершина " << i + 1 << " - доминирующая" << std::endl;

}

std::cout << "\n=========================================================================================\n";

std::cout << "Матрица инцидентности:\n\n";

int\*\* arr1 = new int\* [size];

for (int i = 0; i < size; ++i) {

arr1[i] = new int[count\_smez] {};

}

int d = 0;

for (int i = 0; i < size; ++i) {

for (int j = 0; j < size; ++j) {

if (arr[i][j] == 1 && i < j) {

arr1[i][d] = 1;

arr1[j][d] = 1;

d++;

}

}

}

int count\_in = 0;

for (int i = 0; i < size; ++i) {

for (int j = 0; j < count\_smez; ++j) {

std::cout << arr1[i][j] << "\t";

if (j + 1 > count\_in) {

count\_in = j + 1;

}

}

std::cout << std::endl;

}

std::cout << "\nКоличество рёбер (размер) графа через матрицу инцидентности: " << count\_in << std::endl;

for (int i = 0; i < size; ++i) {

int g = 0;

for (int j = 0; j < count\_smez; ++j) {

if (arr1[i][j] == 1) {

g++;

}

}

if (g == 0) std::cout << "Вершина " << i + 1 << " - изолированная" << std::endl;

if (g == 1) std::cout << "Вершина " << i + 1 << " - концевая" << std::endl;

if (g == size - 1) std::cout << "Вершина " << i + 1 << " - доминирующая" << std::endl;

}

for (int i = 0; i < size; ++i) {

delete[] arr1[i];

}

delete[] arr1;

for (int i = 0; i < size; ++i) {

delete[] arr[i];

}

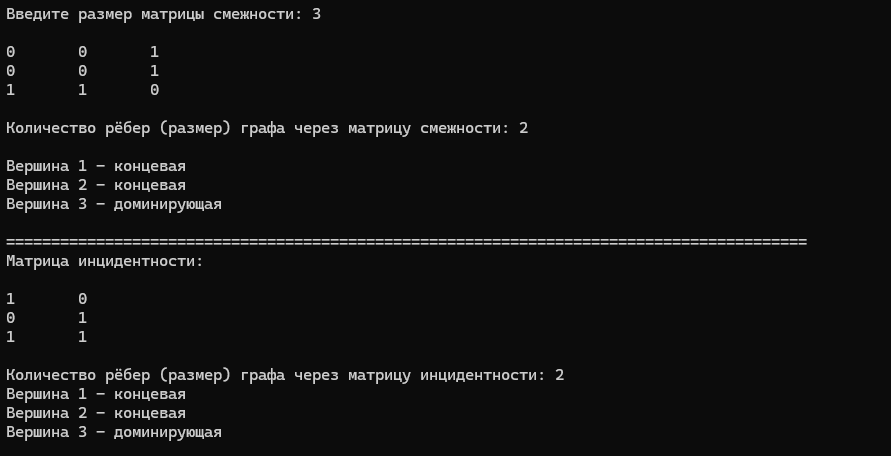
delete[] arr;

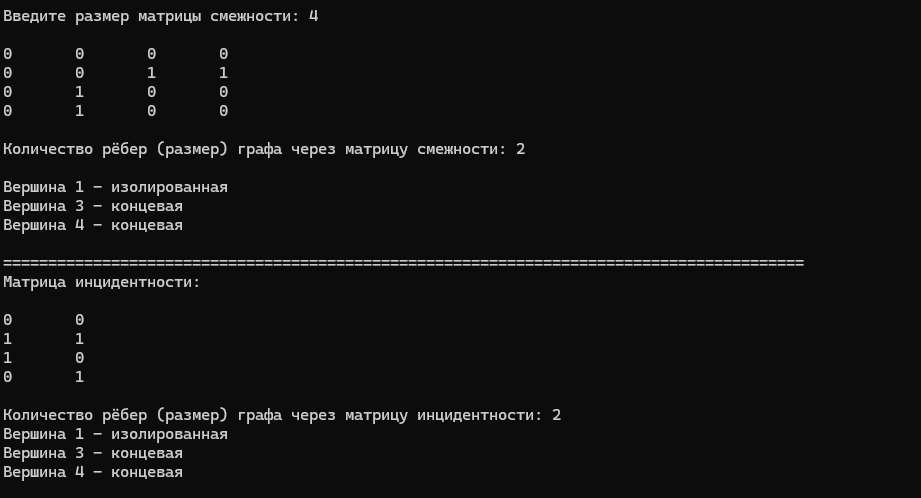
return 0;

}

**Результаты работы программы**

Результаты работы программы представлены на рисунках:

Рисунок 1 - Размер матрицы 3

Рисунок 2 - Размер матрицы 4

**Вывод**

В ходе выполнения лабораторной работы была разработана программа, для определения характеристик графа. Результаты работы программы совпали с ожидаемыми результатами, следовательно, программа работает без ошибок. Также был получен опыт в создании проектов в среде Microsoft Visual Studio и приобретен навык программирования алгоритмов.