Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 5

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Определение характеристик графов»

Выполнили:

студенты группы 22ВВВ2:

Зубриянова А.А.

Кондратьева В.И.

Приняли:

Акифьев И.В.

Митрохин М.А.

Пенза 2023

**Название**

Определение характеристик графа.

**Цель работы**

Научиться определять характеристики 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**

Задание1:

Сгенерировали (используя генератор случайных чисел) матрицу смежности для неориентированного графа. Главную диагональ матрицы заполнили нулями. Так как граф неориентированный – матрица симметрична относительно главной диагонали. Определили размер графа путём подсчета «1» выше главной диагонали. Нашли изолированные, концевые и доминирующие вершины, рассматривая каждую строку матрицы смежности поочередно.

Задание 2:

Построили для графа матрицу инцидентности. Определили размер графа, используя матрицу инцидентности. Нашли изолированные, концевые и доминирующие вершины, рассматривая каждую строку матрицы смежности поочередно.

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <windows.h>

#include <time.h>

#include <stdlib.h>

#include <locale.h>

int main() {

srand(time(NULL));

setlocale(LC\_ALL, "Russian");

int size;

int\*\* arr;

printf("Введите размер матрицы смежности: ");

scanf("%d", &size);

printf("\n");

arr = (int\*\*)malloc(sizeof(int\*) \* size);

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

arr[i] = (int\*)malloc(sizeof(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];

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

//count\_smez++;

}

}

else {

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

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

count\_smez++;

}

}

}

}

//count\_smez = count\_smez / 2;

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

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

printf("%7d", arr[i][j]);

}

printf("\n");

}

printf("\n");

printf("Количество рёбер (размер) графа через матрицу смежности: %d\n", count\_smez);

printf("\n");

int g = 0;

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

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

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

g++;

}

}

if (g == 0) {

printf("Вершина %d - изолирована\n", i + 1);

}

if (g == 1) {

printf("Вершина %d - концевая\n", i + 1);

}

if (g == size - 1) {

printf("Вершина %d - доминирующая\n", i + 1);

}

g = 0;

}

printf("\n=========================================================================================\n");

//задание 2\*

printf("Матрица инцидентности:\n\n");

int\*\* arr1;

arr1 = (int\*\*)malloc(sizeof(int\*) \* size);

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

arr1[i] = (int\*)malloc(sizeof(int) \* count\_smez);

}

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

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

arr1[i][j] = 0;

}

}

int d = 0;

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

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

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

arr1[i][d] = 1;

arr1[j][d] = 1;

d++;

}

}

}

int count\_in = 0; //==count\_smez

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

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

printf("%7d", arr1[i][j]);

if (j + 1 > count\_in) { //размер графа = количеству столбцов матрицы инцидентности

count\_in = j + 1; //на 2 делить не надо

}

//if (arr1[i][j] == 1) { //размер графа = количество единиц деленное на 2

// count\_in++;

//}

}

printf("\n");

}

//count\_in = count\_in / 2;

printf("\n");

printf("Количество рёбер (размер) графа через матрицу инцидентности: %d\n", count\_in);

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

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

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

g++;

}

}

if (g == 0) {

printf("Вершина %d - изолирована\n", i + 1);

}

if (g == 1) {

printf("Вершина %d - концевая\n", i + 1);

}

if (g == size - 1) {

printf("Вершина %d - доминирующая\n", i + 1);

}

g = 0;

}

for (int i = 0; i < size; ++i) free(arr1[i]);

free(arr1);

for (int i = 0; i < size; ++i) free(arr[i]);

free(arr);

return 0;

}

**Результаты работы программы**

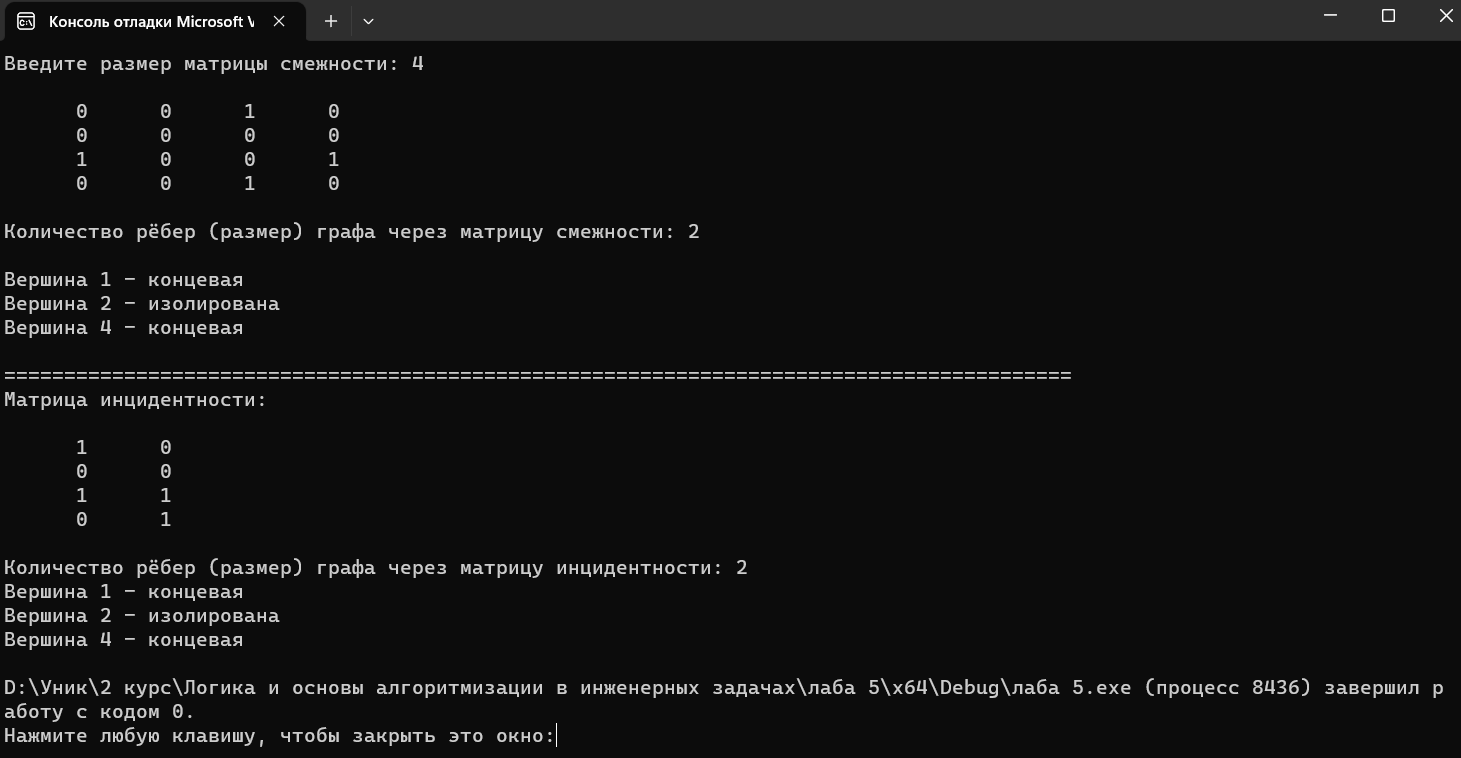


Рисунок 1

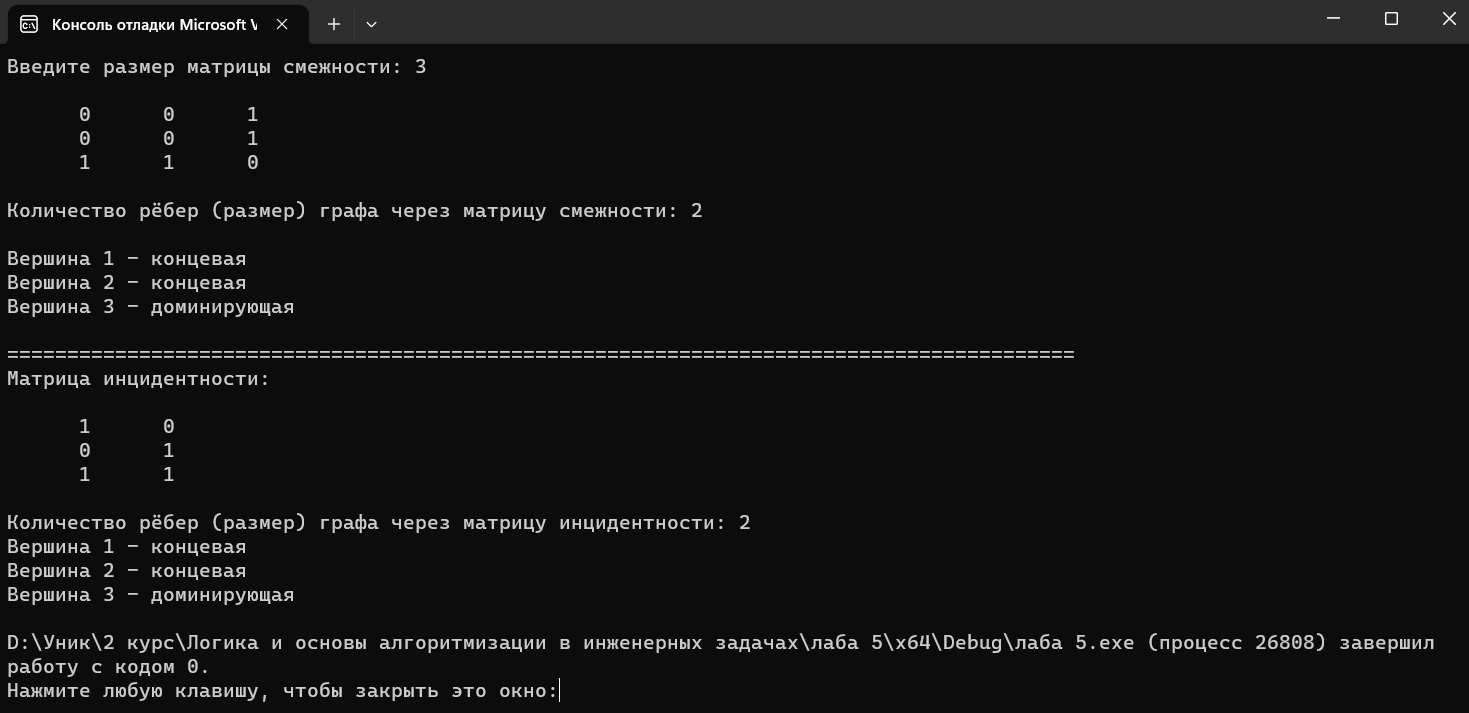


Рисунок 2

**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.

Получили опыт в создании проектов в среде Microsoft Visual Studio, приобрели навыки программирования алгоритмов.