Министерство науки и высшего 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 24ВВВ1

Куничкина В.А.

Суркова Д.А.

**Приняли:**

к.т.н. Деев М.В.

к.т.н. доцент Юрова О.В.

Пенза 2025

Цель работы: Сгенерировать (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран. Определить размер графа *G*, используя матрицу смежности графа. Найти изолированные, концевые и доминирующие вершины.

**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.

**Практическая часть:**

**Листинг:**

#include <conio.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <windows.h>

int main(void) {

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

int n, size = 0;

printf("введите количество вершин\n");

scanf\_s("%d", &n);

int\*\* G = (int\*\*)malloc(n \* sizeof(int\*));

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

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

}

int\* deg = (int\*)malloc(n \* sizeof(int));

int\* loop = (int\*)malloc(n \* sizeof(int));

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

deg[i] = 0;

loop[i] = 0;

}

srand(time(NULL));

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

for (int j = i; j < n; j++) {

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

size += G[i][j];

}

}

printf("Матрица G (%dx%d):\n", n, n);

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

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

printf("%4d ", G[i][j]);

}

printf("\n");

}

printf("Размер: %d\n", size);

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

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

if (i == j) {

loop[i] = G[i][j];

}

else

{

deg[i] += G[i][j];

}

}

deg[i] += 2 \* loop[i];

}

int k1 = 0;

int k2 = 0;

int k3 = 0;

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

if ((deg[i] == 0 && loop[i] == 0) || (deg[i] == 2 && loop[i] == 1)) {

printf("Вершина %d изолированная", i);

k1++;

}

if (deg[i] == 1) {

printf("Вершина %d концевая", i);

k2++;

}

if ((deg[i] - 2 \* loop[i]) == n - 1) {

printf("Вершина %d доминирующая", i);

k3++;

}

}

printf(" Изолированных вершин: %d\n", k1);

printf(" Концевых вершин: %d\n", k2);

printf(" Доминирующих вершин: %d\n", k3);

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

free(G[i]);

}

free(G);

free(deg);

free(loop);

\_getch();

return 0;

}

**Вывод**

В ходе выполнения лабораторной работы были изучены графы и их характеристики инцидентности, которые можно определить по в языке Си