Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №8

по курсу «Логика и основы алгоритмизации в инженерных задачах»

## на тему «Определение характеристик графа»

Выполнил:

студенты группы 20ВВ2

Зиновьев Я.М.

Аверочкин П.С.

Приняли:

к. т. н. доцент Юрова О.В.

д. т. н., профессор Митрохин М.А.

Пенза 2021

**Цель работы**

Определить характеристики взвешенного неориентированного графа.

**Лабораторное задание**

### **Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного взвешенного графа G. Выведите матрицу на экран.

2. Определите радиус и диаметр графа G, используя матрицу смежности графа.

3. Определите подмножества периферийных и центральных вершин графа G, используя матрицу смежности.

4. Найдите изолированные, концевые и доминирующие вершины.

**Описание программы:**

Программа осуществляет поиск эксцентриситета, радиуса и диаметра графа через обход в ширину.  
Так же осуществляется поиск изолированных, концевых и доминирующих вершин.

### **Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <iostream>

#include <stdlib.h>

#include <queue>

#include <time.h>

using namespace std;

void bfsd(int\*\* mas1, int n, int\* dist, int v)

{

    queue<int> q;

    q.push(v);

    dist[v] = 0;

    while (!q.empty())

    {

        v = q.front();

        q.pop();

        for (int i = 0; i < n; i++)

        {

            if (mas1[v][i] > 0 && dist[i] > dist[v] + mas1[v][i])

            {

                q.push(i);

                dist[i] = dist[v] + mas1[v][i];

            }

        }

    }

}

void main()

{

    setlocale(LC\_ALL, "RUS");

    srand(time(NULL));

    int n, num, v; //N-кол-во вершин, num-вершина для ввода, v-вершина для ввода

    int\*\* mas1, \*\* mas2;

    printf("\nУкажите размер матрицы: ");

    scanf\_s("%d", &n);

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

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

    int rad, diam;

    diam = -1;

    rad = 11;

    mas1 = (int\*\*)malloc(n \* sizeof(int\*));

    for (int i = 0; i < n; i++)

    {

        mas1[i] = (int\*)malloc(n \* sizeof(int));

    }

    for (int i = 0; i < n; i++)

    {

        mas1[i][i] = 0;

        for (int j = i + 1; j < n; j++)

        {

            mas1[i][j] = rand() % 11;

            mas1[j][i] = mas1[i][j];

        }

    }

    printf("\n\nВывод матрицы смежности:\n ");

    for (int i = 0; i < n; i++) { // вывод матриц смежности

        printf("\n");

        for (int j = 0; j < n; j++) {

            printf("%d ", mas1[i][j]);

        }

    }

    for (int i = 0; i < n; i++)

    {

        dist[i] = 1000;

    }

    for (int i = 0; i < n; i++)

    {

        ecst[i] = -1;

    }

    for (int i = 0; i < n; i++) {

        bfsd(mas1, n, dist, i);

        printf("\nРасстояние от вершины %d до остальных:\n", i + 1);

        for (int j = 0; j < n; j++)

        {

            printf("%d ", dist[j]);

            if (dist[j] > ecst[i]) {

                ecst[i] = dist[j];

            }

        }

        for (int k = 0; k < n; k++)

        {

            dist[k] = 1000;

        }

    }

    printf("\nЭксцентриитеты\n");

    for (int i = 0; i < n; i++) {

        printf("%d ", ecst[i]);

    }

    for (int i = 0; i < n; i++) {

        if (diam < ecst[i]) {

            diam = ecst[i];

        }

    }

    for (int i = 0; i < n; i++) {

        if (rad > ecst[i]) {

            rad = ecst[i];

        }

    }

    printf("\nДиаметр: %d \n", diam);

    printf("Радиус: %d \n", rad);

    printf("Центральные вершины: \n");

    for (int i = 0; i < n; i++) {

        if (rad == ecst[i]) {

            printf("%d ", i + 1);

        }

    }

    printf("\nПериферийный вершины: \n");

    for (int i = 0; i < n; i++) {

        if (diam == ecst[i]) {

            printf("%d ", i + 1);

        }

    }

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

    for (int i = 0; i < n; i++) {

        deg[i] = 0;

    }

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            if (mas1[i][j] != 0) {

                deg[i] = deg[i] + 1;

            }

        }

    }

    printf("\nВершины ");

    for (int i = 0; i < n; i++) {

        if (deg[i] == 0) {

            printf("%d", i + 1);

        }

    }

    printf(" - изолированные");

    printf("\nВершины ");

    for (int i = 0; i < n; i++) {

        if (deg[i] == 1) {

            printf("%d", i + 1);

        }

    }

    printf(" - концевые");

    printf("\nВершины ");

    for (int i = 0; i < n; i++) {

        if (deg[i] == (n - 1)) {

            printf("%d ", i + 1);

        }

    }

    printf(" - доминирующие");

}

**Результаты работы программы**

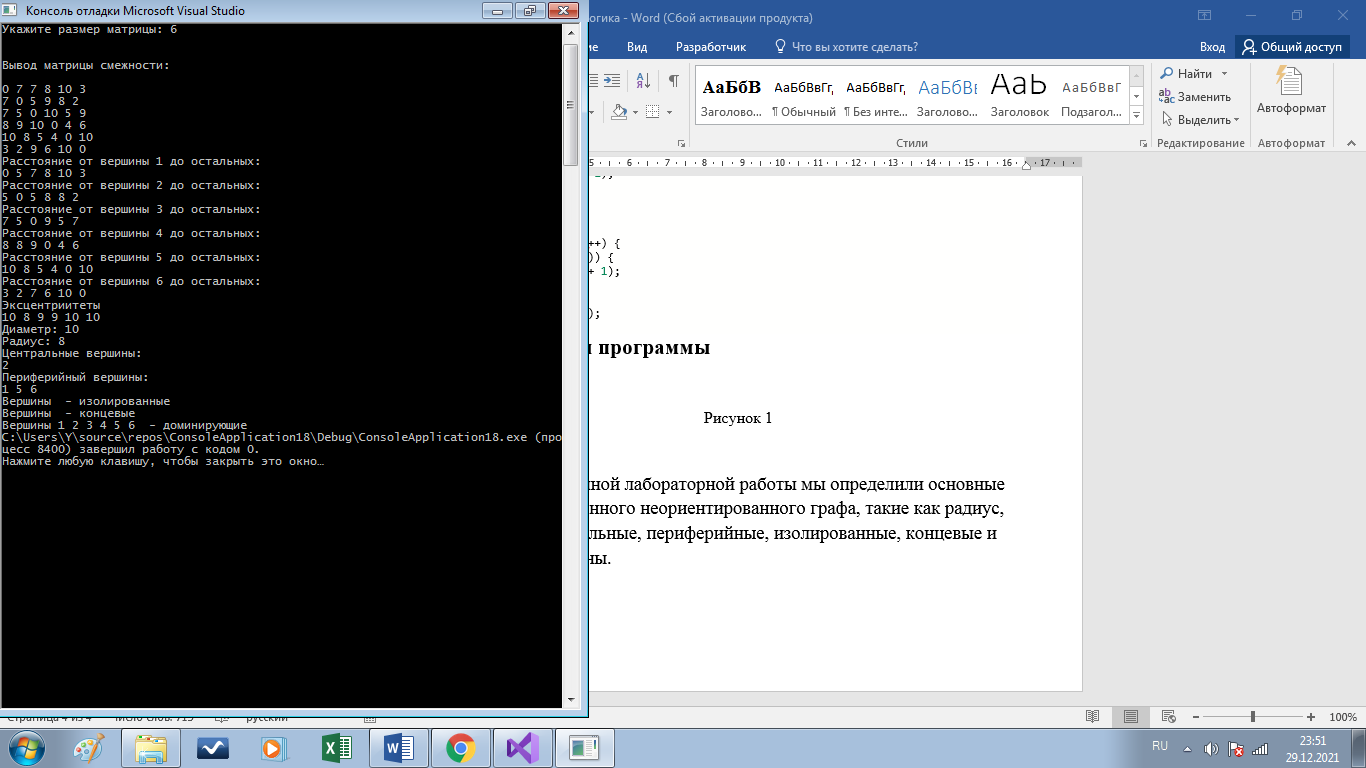


Рисунок 1

## Вывод

В ходе выполнения данной лабораторной работы мы определили основные характеристики взвешенного неориентированного графа, такие как радиус, диаметр, нашли центральные, периферийные, изолированные, концевые и доминирующие вершины.